ELETTRONICA

LA RIVISTA DI ELETTRONICA APPLICATA, TECNOLOGIE E PRODOTTI

MAGGIO 2005 - ANNO 21 - N° 239

€ 5,50

Poste Italiane Spa - Spedizione in abbonamento Postale - D.L. 353/2003 (conv. in L. 27/02/2004 n. 46) art. 1, comma1, DCB Milano.
In caso di mancato recapito, restituire all'editore che si impegna a pagare la relativa tassa presso il CPM di Roserio - Milano

COSTRUIRE UN GENERATORE EOLICO PARTENDO DA ZERO

SIMCopier
Duplicatore di rubrica telefonica

ELETTRONICANDO
Una panoramica sui diodi

PRATICA
MICROBASIC
Le variabili, le costanti e gli ingressi

Guida pratica alla fotoincisione dei PCB

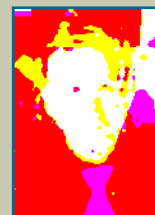
TEORIA
INTELLIGENZA ARTIFICIALE
Implementazione delle reti neurali

ALIMENTATORI SWITCHING
I trasformatori per gli switching

SPECIALE
Filtro di Kalman e sue applicazioni

www.farelettronica.com

INWARE EDIZIONI

Tiziano Galizia (t.galizia@farelettronica.com)

Un anno ricco di novità

Prosegue l'evoluzione di Fare Elettronica, con alcuni cambiamenti mirati ad assecondare sempre più le esigenze dei nostri lettori e di un mercato editoriale che cambia in fretta.
Abbiamo iniziato con il nuovo layout che utilizza tratti grafici moderni, un'impaginazione rigorosa ed ordinata ed una struttura editoriale più completa. Adesso è arrivato il tempo di attuare l'ultima parte di un'evoluzione che ci è costata fatica ed investimenti ma che, come i fatti dimostrano, sta portando risultati di rilievo. Sto parlando nel nuovo piano di uscite annuale e del nuovo logo.
Partiamo dal nuovo piano di uscite. Fino a questo numero Fare Elettronica usciva in edicola con undici numeri dei quali uno doppio (Luglio/Agosto) che, a fronte di un prezzo di copertina più alto (€ 6,00), offriva trentadue pagine in più. Questo piano cambia a partire dal prossimo numero, infatti, Fare Elettronica sarà in edicola sempre con undici numeri ma non ci sarà più il numero doppio.
Gli undici numeri potranno avere sedici, trentadue o anche più pagine a seconda dei contenuti, ultimamente la dimensione degli articoli sta infatti mettendo in crisi la redazione, rendendo il vecchio telaio "a scatola fissa" inadatto e obsoleto. In effetti, già lo scorso numero di aprile e quello che avete in mano adesso hanno entrambi sedici pagine in più, operazione che ha dato un rinnovato respiro e un aggiunto valore per i nostri lettori.
Durante il corso dell'anno ci saranno anche due CD omaggio allegati alla rivista e non prevediamo alcun ulteriore incremento sul prezzo di copertina, normalmente giustificato dalla presenza del CD. Nell'ottica di questa rinnovata strategia editoriale, si è reso necessario un adeguamento del prezzo di copertina, che diventa da questo numero € 5,50, mentre il prezzo dell'abbonamento è già cambiato dal numero scorso: € 45,00. Come abbiamo sempre ripetuto i vantaggi per i nostri abbonati restano, per esempio, la garanzia del prezzo bloccato fa si che queste novità non abbiano alcun impatto economico, ma sono di fatto un valore aggiunto che premia la preziosa fedeltà. Vi consiglio quindi di abbonarvi e di farlo subito, per non perdere neanche una delle nuove iniziative.
Veniamo adesso al nuovo logo che porta con se un nuovo design della copertina, già a partire dal prossimo numero di Giugno. Ecco quindi in anteprima il nuovo logo che troverete il prossimo mese in edicola!

Vorrei adesso soffermarmi su quelli che sono i contenuti di Fare Elettronica. Moltissimi lettori hanno apprezzato il nuovo piano editoriale definendolo "innovativo", mentre alcuni hanno lamentato una diminuzione del numero di progetti pratici, lamentela che non è caduta nel vuoto, infatti, già da questo numero potete notare due nuovi progetti all'avanguardia: il generatore eolico ed il copiatore di SIM; ma questo è solo l'inizio, molti altri progetti sono in arrivo, tutti di grande interesse.
Anche sul fronte dei corsi stiamo lavorando senza sosta, ecco alcune anticipazioni: un corso teorico/pratico sulle CPLD, uno sulle tecniche DDS (Direct Digital Synthesis) con i PICmicro, uno sui microprocessori ATMEL AVR e tanto altro ancora.

Non mi resta che augurarvi una piacevole lettura e rinnovarvi l'appuntamento in edicola a Giugno con un nuovo e bellissimo numero di Fare Elettronica.

DIRETTORE RESPONSABILE
Antonio Cirella

DIRETTORE DI REDAZIONE
Tiziano Galizia

DIRETTORE TECNICO
Maurizio Del Corso

HANNO COLLABORATO IN QUESTO NUMERO
Giovanni Di Maria, Riccardo Nicoletti, Paolo D'Angella, Antonio Di Stefano, Giuseppe Modugno, Agostino Rolando, Romano Bernarducci, Nico Grilloni, Antonio Cecere.

DIREZIONE • REDAZIONE • PUBBLICITÁ
INWARE srl - Via Cadorna, 27/31 - 20032 Cormano (MI)
Tel. 02.66504794 - 02.66504755 - Fax 02.66508225
info@inware.it - www.inwaredizioni.it

PROGETTO GRAFICO E IMPAGINAZIONE
Graficonsult - Milano

STAMPA
ROTO 2000 - Via L. da Vinci, 18/20 - 20080, Casarile (MI)

DISTRIBUZIONE
Parrini & C. S.p.a. - Viale Forlanini, 23 - 20134, Milano

UFFICIO ABBONAMENTI
PARRINI & C. S.p.a.
Servizio abbonamenti
Viale Forlanini, 23 - 20134 Milano
Per informazioni, sottoscrizione o rinnovo dell'abbonamento:
abbonamenti@farelettronica.com
Tel. 02.66504794 - Fax. 02.66508225

Poste Italiane S.p.a.
Spedizione in abbonamento Postale
D.L. 353/2003 (conv. In L. 27/02/2004 n. 46) art. 1, comma1, DCB Milano.

Abbonamento per l'Italia: € 45,00
Abbonamento per l'estero: € 115,00

Gli arretrati potranno essere richiesti, per iscritto, al seguente costo:
Numero singolo: € 7,50
Numero doppio: € 9,00
Numero con allegato: € 8,50

Autorizzazione alla pubblicazione del Tribunale di Milano n. 647 del 17/11/2003.
Iscrizione al R.O.C. n. 11035 19/11/2003

RICHIESTE DI ASSISTENZA
Per richiedere assistenza o chiarimenti sugli articoli pubblicati, vi preghiamo di contattare direttamente l'autore. Se questo non fosse possibile, utilizzate il modulo di contatto che trovate sul nostro sito web www.farelettronica.com.

COLLABORARE CON FARE ELETTRONICA
Le richieste di collaborazione vanno indirizzate all'attenzione di Tiziano Galizia *(t.galizia@farelettronica.com)* e accompagnate, se possibile, da una breve descrizione delle vostre competenze tecniche e/o editoriali, oltre che da un elenco degli argomenti e/o progetti che desiderate proporre.

COME CONTATTARCI
INWARE Edizioni
Via Cadorna, 27/31 - 20032 Cormano (MI)
Telefono 02.66504794 - Fax 02.66508225
Redazione: redazione@farelettronica.com
Ufficio Abbonamenti: abbonamenti@farelettronica.com

ELENCO INSERZIONISTI

Pratica

pag. 50

Teoria

pag. 16

Risorse

pag. 28

Speciale

239
Maggio
2005

Scrivete a:
MAILBOX
REDAZIONE DI
FARE ELETTRONICA
Inware s.r.l.
Via Cadorna, 27/31
20032, Cormano (MI)

Oppure inviate un'email a:
mailbox@farelettronica.com

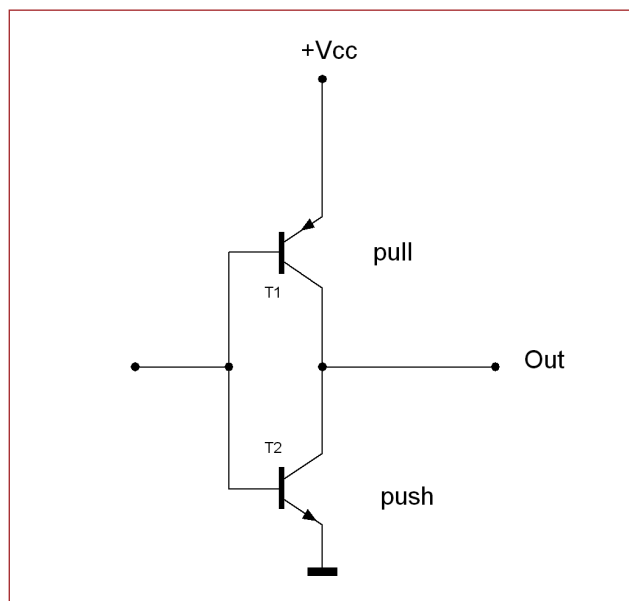
Mailbox

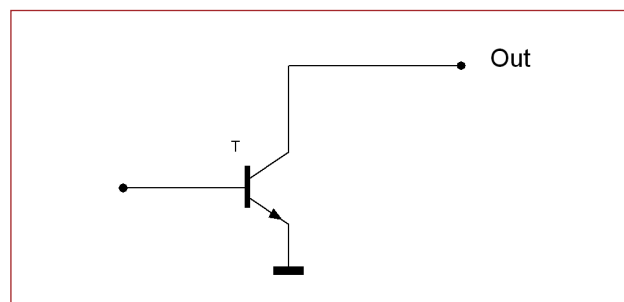
Questa rubrica ospita le richieste più interessanti pervenute dai lettori. Per quanto possibile verrà data risposta a tutte le richieste pervenute via email.

STADI DI USCITA

Mi sono recentemente imbattuto, studiando i microcontrollori ST6, in alcuni termini tecnici (riscontrati anche in alcuni vostri articoli sui microcontrollori serie PIC) di cui non conosco l'esatto significato. Vorrei sapere qual è la differenza tra un'uscita push-pull ed un'uscita open-collector a livello fisico e applicativo.

Simone Milli

Lo stadio di uscita **push-pull** può essere schematizzato con il circuito nella figura seguente:

In altre parole esiste sempre una coppia di transistori (possono essere BJT o FET) in cui uno si trova sempre in saturazione e l'altro in interdizione o viceversa. Questa configurazione garantisce che il valore dell'uscita sia sempre ben definito. Infatti se alla base dei due transistori viene applicato un livello alto di tensione T1 si interdice mentre T2 va in saturazione portando l'uscita a livello basso (effetto Push). Viceversa se alla base dei transistori viene applicato un livello basso di tensione, sarà T1 in saturazione e T2 interdetto, quindi l'uscita sarà a livello alto (+Vcc). La presenza dei due transistori garantisce l'erogazione di una elevata corrente in uscita in caso di uscita alta (grazie a T1) e l'assorbimento di alte correnti dall'uscita in caso di uscita bassa (grazie a T2). Questa configurazione è nota anche come "totem-pole".

Uno stadio di uscita **open collector** può essere schematizzato nel seguente modo:

Ovviamente se viene utilizzato un transistore FET anziché un BJT, la configurazione prende il nome di Open Drain. Si noti che in questo caso manca del tutto il transistore responsabile del collegamento dell'uscita a +Vcc. Ciò significa che l'uscita è in grado solo di assorbire grandi correnti sul livello basso, ovvero quando alla base del transistore viene applicato un livello alto di tensione. In questo caso il problema nasce se alla base del

“Richieste, chiarimenti, dubbi e commenti dai lettori”

transistore viene applicato un livello basso di tensione. In questo caso T è interdetto e l’uscita risulta essere flottante (scollegata sia dalla massa che dal positivo di alimentazione).
Per evitare questo stato indefinito dell’uscita è necessario inserire una resistenza esterna (resistenza di pull-up) tra l’uscita ed il positivo di alimentazione. In questo caso quando il transistore è interdetto (quindi come se fosse un circuito aperto) l’uscita viene mantenuta a livello alto dalla resistenza di pull-up. Il livello basso è invece garantito dalla saturazione del transistor.

CANCELLAZIONE UV

Possiedo una lampada UVC da 6 W con la quale devo cancellare un PIC16C745/JW: a quale distanza dal PIC devo posizionare la lampada e per quanto tempo devo esporre il PIC alla stessa?

Lorenzo Ghirelli

Per la cancellazione di dispositivi in tecnologia EPROM siano essi microcontrollori che memorie, è necessario disporre di una lampada UV in grado di emettere luce ultravioletta della lunghezza d’onda di 253,7nm (2537 Å). Lampade di questo tipo sono quelle usate per la sterilizzazione (l’avrete sicuramente viste dal parrucchiere o dal dentista).
In alternativa è possibile utilizzare anche le lampade abbronzanti, purché si ponga attenzione a non surriscaldare troppo il chip per non danneggiarlo, o anche le lampade per bromografi a condizione che la lunghezza d’onda della luce emessa non sia troppo lontana dal valore richiesto. Non sono adatte invece le lampade a luce fredda ovvero quelle comunemente usate nelle discoteche per esaltare il bianco ed i materiali fluorescenti.
La lampada deve essere posta a circa due centimetri dal chip ed il tempo di esposizione dipende dal tipo di lampada usata. Per determinare il giusto tempo di esposizione si consiglia di iniziare con due minuti, quindi verificare se il chip è stato effettivamente cancellato. Qualora il chip non sia totalmente cancellato, si deve incrementare il tempo di esposizione di un minuto, quindi effettuare nuovamente la verifica. Quando il chip risulta cancellato, il giusto tempo di esposizione si ottiene moltiplicando per 1,5 l’ultimo tempo di esposizione. Ad esempio se dalle prove effettuate risulta che il chip viene cancellato in 5 minuti, il giusto tempo di esposizione sarà di 1,5x5=7,5 minuti.
Questo garantisce che il chip venga cancellato correttamente evitando il rischio di una cancellazione leggera (che lascerebbe traccia dei dati precedenti) e di una cancellazione pesante che danneggerebbe il chip impedendone la successiva riprogrammazione.
Le lampade UV sono soggette ad invecchiamento per cui via via che vengono utilizzate, il tempo di esposizione deve necessariamente aumentare. Sarà dunque opportuno ricalcolare il tempo di esposizione dopo un centinaio di operazioni di cancellazione.

CD ALLEGATO

Il CD allegato all’ultimo numero di Fare Elettronica (Aprile 2005), non serve a molto visto che è una versione limitata. Molte persone compreso me non hanno gradito il fatto che l’avete fatto pagare sul costo della rivista.

Antonio Rossi

Siamo spiacenti di constatare che i nostri sforzi per avere una versione in italiano del famoso CD ECA non siano, nel suo caso, considerati interessanti. Per converso, molti nostri lettori trovano utile l'accesso al database completo (150mila

semiconduttori "in linea" non sono proprio inutili) e molti hanno trovato ottimo lo sconto del 20% per l'acquisto della versione completa.
Grazie comunque del tempo che ci ha dedicato per comunicarci le sue (per noi preziose) impressioni, e speriamo che troverà in futuro sempre un motivo valido per leggere la nostra rivista, frutto di molti sforzi da parte nostra, volti alla soddisfazione della maggior parte dei lettori.

CONGRATULAZIONI

Finalmente dopo tanti anni di assenza dallo scenario editoriale nostrano, è arrivato qualcosa di innovativo che sa coniugare la pratica con la giusta rigorosità teorica. Penso che la via che state seguendo sia quella vincente, finalmente non più la solita scatola di montaggio accompagnata dal solito articolo letto magari con rassegnata pigrizia mentale. Nei vostri articoli, invece, il lettore si trova coinvolto dinamicamente, ritrova la gioia di imparare, approfondire e sperimentare. Ben tornata Fare Elettronica.

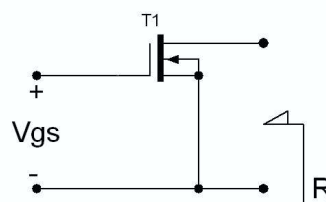
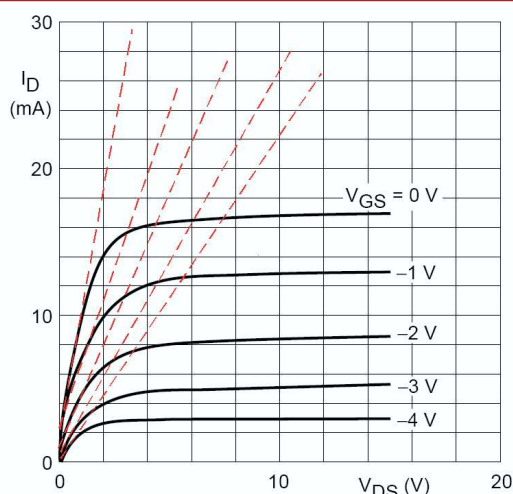
Gianni Risi

La ringraziamo moltissimo per i suoi complimenti al nostro lavoro, rafforzano l'idea che "la rivista che abbiamo in mente" stia andando per la giusta strada, la rivista che da lettori avremmo voluto leggere. Come lei non siamo sperimentatori passivi ma curiosi che vogliono capire il perchè delle cose, "sperimentatori" e non "smanettoni". Fa sempre piacere quando qualcuno coglie il senso di ciò che stiamo facendo: una "Rivista di Cultura Elettronica". Spero che Lei sia un nostro abbonato e che ci si possa incontrare personalmente ad uno dei prossimi appuntamenti (www.farelettronica.com/fiere.asp).

POTENZIOMETRO ELETTRONICO

Avrei la necessità di disporre di una resistenza il cui valore sia variabile e controllabile con una tensione continua. Questo mi permetterebbe di sostituire un potenziometro tradizionale con uno di tipo "elettronico".

Luca Chini

Il modo più semplice di ottenere una resistenza controllata in tensione, è quello di utilizzare un transistore ad effetto di campo, quindi un MOSFET o un JFET polarizzato in zona triodo. Osservando infatti le caratteristiche di uscita di un transistore FET in questa zona (riportate nella figura seguente), si nota come la loro pendenza sia dipendente dalla tensione Vgs applicata tra gate e source. La pendenza delle caratteristiche è inversamente proporzionale alla resistenza quindi la linea a pendenza maggiore corrisponde ad una resistenza minore. La resistenza è quella che si misura tra source e drain.

ERRATA CORRIGE

Ci scusiamo con i lettori per l'errore commesso nella pubblicazione della pubblicità "Mikrobasic" (FE 238 - Aprile 2005 - pag. 63), la promozione ivi indicata non era valida.
Quella corretta è stata pubblicata in questo numero (pag. 53).

Rubrica di notizie e novità dal mondo dell'elettronica.

Prima pagina

239-01 EEPROM SERIALI AD ALTA DENSITÀ IN PACKAGE ULTRA-SMALL DFN DA 2X3MM

Microchip introduce una nuova opzione di package DFN (dual flat no leads) da 2x3 millimetri destinata ai prodotti della sua linea di EEPROM seriali I2C™ e Microwire®. La nuova opzione garantisce gli ingombri estremamente ridotti e i bassi profili richiesti dalle applicazioni vincolate dal punto di vista degli spazi, quali i dispositivi portatili e i computer laptop. Le nuove EEPROM seriali I2C da 64 Kbit e Microwire da 16 Kbit in package DFN offrono le soluzioni a più alta densità alloggiate nel package standard più piccolo attualmente in commercio.

Con un ingombro su scheda di soli 6 millimetri quadri, il nuovo package DFN è più piccolo dell'80% rispetto a un package SOIC 8-lead; il 59% più piccolo rispetto a un package MSOP 8-lead; e il 27% più piccolo rispetto a un package SOT-23 5-lead. Oltre a questo, alle applicazioni dove è richiesto un profilo estremamente basso, il package DFN offre uno spessore di soli 0.9 millimetri, tra il 20 e 40% in meno rispetto agli altri package standard.

Il nuovo package DFN è disponibile per la linea 24xx di EEPROM seriali I2C con densità da 128 bit a 64 Kbit, e per la linea 93xx di EEPROM seriali Microwire con densità da 1 Kbit a 16 Kbit. Il package DFN da 2x3 millimetri è immune da piombo (Pb) ed è compatibile con i processi di saldatura a riflusso lead-free ad alta temperatura e con i processi stagno/piombo esistenti

239-02 SOLUZIONI PCI EXPRESS

IIDT™ (Integrated Device Technology, Inc.) introduce la nuova famiglia PRECISE™ di soluzioni PCI Express™, costitutia da quattro nuovi prodotti di bridging e switching. Nati per le applicazioni server e storage ad alte prestazioni, i nuovi dispositivi IDT sono i primi a garantire delle soluzioni economiche di bridging e di espansione di I/O PCI-Express. IDT offrirà ai clienti anche un kit completo di sviluppo hardware/software comprendente schede di valutazione, driver software Linux e Windows based e una GUI (graphical user interface) che permette di gestire le funzioni di configurazione e ottimizzazione del sistema.

IDT offre attualmente due switch PCI Express a 3 porte. La proposta comprende le versioni PES12N3 (12 lane) e PES24N3 (24 lane).

Gli switch PCI Express di IDT offrono un'espansione di I/O che consente di potenziare i dispositivi "north bridge" con ulteriori risorse di connettività di I/O ad alte prestazioni.

La famiglia PRECISE™ prevede anche il modello PEB20N1 (bridge PCI Express to PCI-X 2.0 mode 1) e il modello PEB20N2 (bridge PCI Express to PCI-X 2.0 mode 2). Il modello PEB20N2 è il primo dispositivo di questo tipo e rappresenta una soluzione ideale per le applicazioni server ad alte prestazioni. Questi dispositivi collegano le porte PCI Express ad alte prestazioni alle periferiche di precedente generazione. Ciò permette ai progettisti di tutelare le prestazioni della CPU sfruttando le periferiche commodity dotate di

PER SAPERNE DI PIÙ

Per approfondire le notizie riportate in questa rubrica, visitate il sito **www.farelettronica.com/primapagina** e seguite le istruzioni.
In alternativa potete scrivere a:

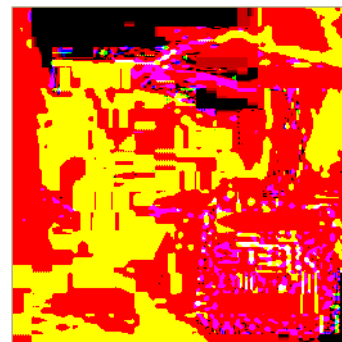
Inware Edizioni
Servizio Prima Pagina
Via Cadorna 27 - 20032 Cormano (MI)

Indicando il codice riportato accanto al titolo della notizia (esempio 239-03).

interfacce correnti. Tale processo è normalmente associato al termine "forward bridging." I dispositivi aiutano anche le attività di "reverse bridging", cioè l'interfacciamento tra porte north bridge dotate interfacce di generazione precedente (tipicamente PCI-X) e le nuove periferiche PCI Express. I prodotti della nuova famiglia di soluzioni bridging e switching PCI Express sono tutte compatibili con gli standard PCI-X

239-03 TDK SEMICONDUCTOR ANNUNCIA IL NUOVO FAX/MODEM ANALOG FRONT END

TDK Semiconductor Corporation, ha presentato il nuovo nato della famiglia di dispositivi modem Analog Front End (AFE). Il nuovo 73M1903C è ottimizzato per le architetture MFP/Fax emergenti e possiede caratteristiche di conformità PSTN a livello internazionale con le terminazioni di linea programmabili. Il 73M1903C integra diverse caratteristiche precedentemente supportate mediante componenti esterni ed è la soluzione più economica per applicazioni a bassa e alta velocità, tra le quali fax V.17/V.34, MFP, POS e decoder televisivi. La caratteristica veramente unica di questo dispositivo è il supporto integrato delle terminazioni PSTN selezionabili via software.

Questo particolare offre ai progettisti la possibilità di supportare i vari standard internazionali con un un unico circuito DAA trasformatore a basso costo.
Il 73M1903C, basato sulle apprezzate serie di modem analogici di TDK 73M1903 e 73M2901CE, include PLL on-chip, fino a 16KHz di tasso di campionamento e interfacce per processori master/slave o daisy-chain.

239-04 REGOLATORE CC/CC BUCK-BOOST AD ALTA CORRENTE CON SINGOLO INDUTTORE

Linear Technology Corporation presenta l'LTC3780, un potente regolatore a commutazione buck-boost che funziona con tensioni di ingresso superiori, inferiori o uguali alla tensione di uscita. Fino ad oggi i circuiti in modalità buck-boost ad alta potenza dipendevano dai trasformatori o da due convertitori CC/CC, uno utilizzato per la conversione step-up (boost) e l'altro per la conversione step-down (buck). L'LTC3780 utilizza un induttore singolo ed è in grado di fornire corrente in uscita fino a 75W; il raddrizzatore sincrono a 4 switch garantisce rendimenti fino al 97%.
L'architettura a modalità di frequenza e corrente costante permette di ottenere una frequenza phase-lockable fino a 400kHz. Grazie all'ampio range della tensione di ingresso da 4V a 30V (36V max) e di uscita da 0,8V a 30V, nonché al passaggio automatico da una modalità operativa all'altra, l'LTC3780 è la soluzione ideale per il

settore automobilistico, delle telecomunicazioni e dei sistemi a batteria. L'LTC3780 utilizza un'architettura di controllo della corrente proprietaria per il funzionamento a frequenza costante in modalità buck o boost.
Il dispositivo è inoltre dotato di protezione contro i guasti in condizioni di sovratensione, sovracorrente e cortocircuito in tutte le modalità di esercizio. Un pin open drain della corretta uscita di corrente segnala se la tensione di uscita è compresa o meno nell'intervallo di regolazione. L'LTC3780 supporta tre diverse modalità operative: frequenza costante, funzionamento Burst Mode e corrente discontinua. L'LTC3780 è fornito in un package SSOP a 24 pin e QFN 5 x 5 mm.

239-05 LA STMICROELECTRONICS ASSEMBLA FINO A OTTO DISPOSITIVI IN UN SOLO PACKAGE MULTI-CHIP

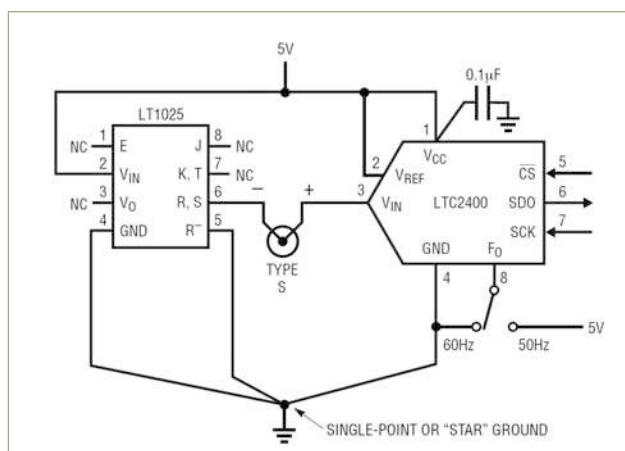
La STMicroelectronics, ha comunicato di essere in grado di produrre package BGA (Ball Grid Array) di soli 1,6 mm di spessore che contengono un massimo di otto chip di memoria uno sopra l'altro. La stessa tecnologia è utilizzata per assemblare due chip di memoria in un package UFBGA (Ultra thin Fine pitch BGA) con uno spessore di soli 0,8mm. I dispositivi prodotti con la nuova tecnologia potranno soddisfare la domanda crescente di memorie sempre più grandi in spazi sempre più piccoli nei telefoni, macchine fotografiche e computer palmari.
I dispositivi MCP (Multi-Chip Package) contenenti da due a quattro chip di memoria spesso differenti tra loro, come SRAM, Flash o DRAM, vengono già ampiamente usati in quei prodotti in cui lo spazio disponibile è ridotto al minimo, come i telefoni cellulari. Sovrapponendo letteralmente due o più chip uno sopra l'altro, si riesce a occupare sulla scheda a circuito stampato lo stesso spazio che sarebbe necessario per un solo chip. Poiché le diverse memorie utilizzano molti segnali comuni, per esempio i bus dei dati e degli indirizzi, il numero di collegamenti tra il package e la scheda rimane praticamente inalterato.
L'utilizzo di chip fisicamente distinti tra loro permette di assemblare memorie di diverso tipo, anche prodotte con tecnologie diverse - ad esempio Flash NOR e SRAM - sfruttando lo stesso spazio fisico nel sistema. Alcune combinazioni tipiche sono, per esempio, Flash NAND e DRAM, o Flash NOR e Flash NAND e SRAM. Il package adatto ad alloggiare due chip potrebbe riunire un ASIC e una Flash NAND, o una Flash NOR e una SRAM.
Fino ad oggi il numero di chip che poteva essere ragionevolmente utilizzato in un package multichip era limitato dallo spessore del singolo die (la piastrina di silicio che costituisce il chip). Le tipiche applicazioni che utilizzano la tecnologia MCP devono rispettare dei limiti di volume totale, oltre che di spazio occupato sulla scheda a circuito stampato. Molte volte non è possibile aumentare lo spessore del package. La nuova tecnologia della ST utilizza dei die molto sottili, spessi solo 40μm. All'interno del package i die vengono sorretti e interspaziati da appositi strati di materiale di "interposizione" ciascuno con uno spessore di 40μm. Il nuovo MCP da 8 chip può alloggiare una "pila" di otto chip attivi di memoria e sette interpositori, mentre i chip del sottilissimo dispositivo UFBGA da due die sono separati da un unico distanziatore.
Uno dei problemi più importanti che sono stati risolti è la tecnologia necessaria per produrre wafer così sottili, con uno spessore pari a un quarto del normale. Inoltre la ST sta utilizzano nuove tecniche per ridurre a circa 40μm l'altezza del percorso del filo di connessione, un filo in oro che collega il die ai pad (piazzole per l'interfacciamento con il mondo esterno) del package.
Quando i chip sono così vicini l'uno all'altro all'interno del package, l'altezza del percorso di questi fili di connessione deve essere decisamente inferiore al normale. La capacità di assemblare memorie di grandi dimensioni in un volume molto ridotto è utile soprattutto nei telefoni cellulari 3G che gestiscono contenuti multimediali, nei palmari più sofisticati e nelle macchine fotografiche digitali. Nello stesso spazio che in precedenza ospitava una memoria da 1 gigabit (1Gb) , per esempio, sarà ora possibile montare 1 gigabyte (1GB) utilizzando un package in cui sono impilati otto chip da 1 Gb.

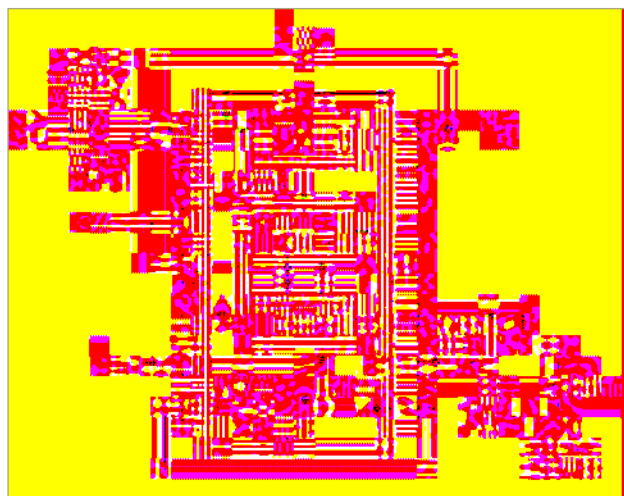
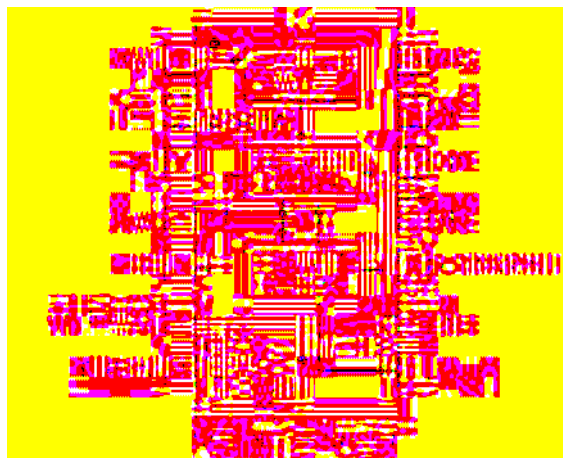
Dal blocco note di Fare Elettronica una raccolta di idee da tenere sempre a portata di mano.

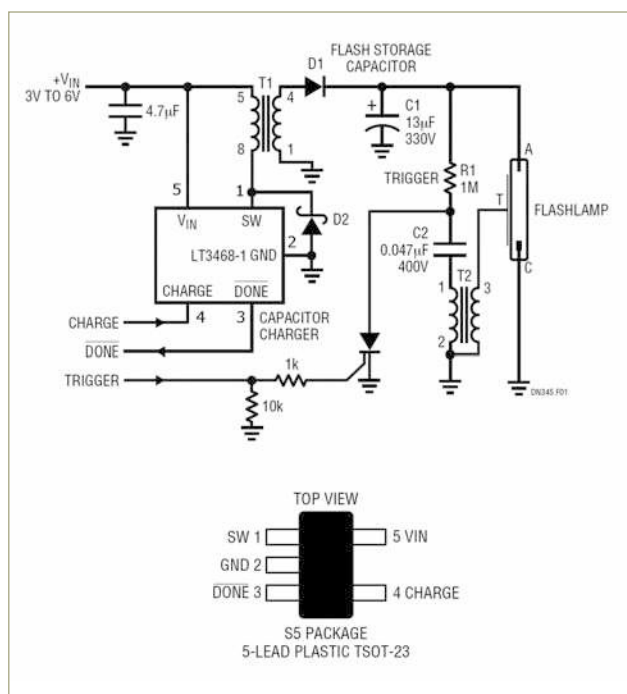
Notepad

Questa rubrica ha lo scopo di fornire degli schemi applicativi o idee di progetto dei componenti elettronici più interessanti, selezionati per voi dalla redazione. Tutti gli schemi presentati sono elaborazioni di quelli ufficiali proposti dai produttori nella documentazione ufficiale.

LETTURA E DIGITALIZZAZIONE DI UNA TERMOCOPPIA

Impiegando un LT1025 ed un LTC2400, è possibile effettuare una lettura da una termocoppia di tipo S . L'LT1025 introduce una tensione di offset in base alla temperatura della giunzione fredda.
L'LCT2400 è un convertitore AD che permette la digitalizzazione della temperatura con una risoluzione di 0.25°C.

CONVERSIONE TRUE RMS-TO-DC

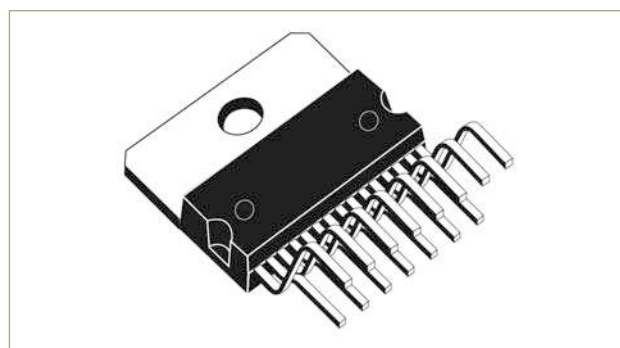
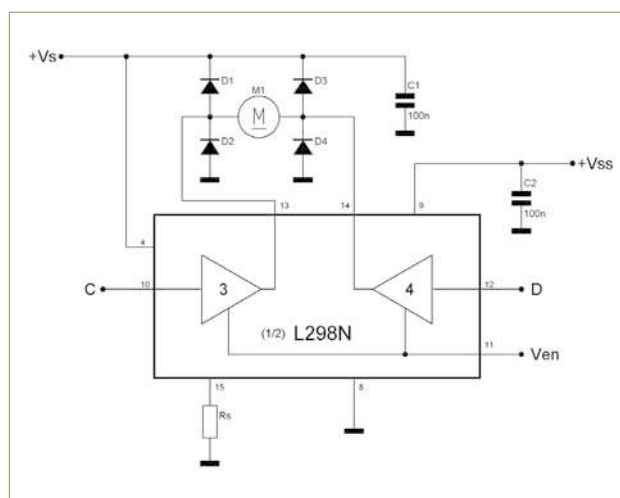
Il chip AD636 di Analog Devices consente la conversione TrueRMS-to-DC in single chip per segnali fino a 200mVrms. Alimentabile con tensioni duali da ±2.5V a ±16.5V o tensioni singole da +5V a +24V, fornisce una tensione di uscita pari alla radice quadrata del valor medio di Vin2.
Mediante trimmer esterni è possibile tarare il fattore di scala e l'offset.
Nelle figure che seguono il pinout ed uno schema applicativo.

CIRCUITO DI CONTROLLO PER LAMPADA FLASH

Per pilotare una lampada flash in fotocamere per telefoni cellulari o simili, è possibile utilizzare il circuito di figura impiegante un LT3468-1 che carica il condensatore C1 fino all'80% della sua carica massima quindi viene innescato il flash mediante un trigger sull'SCR.
Quando il condensatore ha raggiunto la giusta carica ed il circuito è pronto per il flash, viene notificato portando a livello basso il pin DONE. Il pin Charge permette di avviare la carica del condensatore Cf.

DOPPIO DRIVE FULL-BRIDGE PER MOTORI

Per il pilotaggio di un motore DC con controllo della direzione di rotazione e velocità di arresto, è possibile utilizzare lo schema seguente che impiega un L298. A seconda dei livelli dei segnali Ven, C e D viene pilotato il motore in accordo con la tabella riportata di seguito.

Inputs		Function
V_{en} = H	C = H ; D = L	Forward
	C = L ; D = H	Reverse
	C = D	Fast Motor Stop
V_{en} = L	C = X ; D = X	Free Running Motor Stop

L = Low H = High X = Don't care

È opportuno proteggere il dispositivo inserendo i diodi di libera circolazione in parallelo agli avvolgimenti dei motori.

Elettronicando

Il diodo pn rappresenta il dispositivo non lineare più semplice. La sua caratteristica di limitare il passaggio di corrente in un solo verso lo rende l'elemento basilare per il trattamento dei segnali, da cui discendono poi tutti i dispositivi e circuiti non lineari. Presentiamo alcune semplici applicazioni del diodo pn, per capire il funzionamento di alcuni circuiti di base. Ci occuperemo poi di altri tipi di diodi, evidenziando per ciascuno le caratteristiche principali. Numerose simulazioni ci aiuteranno a comprendere il funzionamento dei circuiti descritti.

SEMPLICI APPLICAZIONI DEL DIODO PN

Nel testo ci riferiamo, per semplicità, a diodi ideali. Consideriamo cioè il diodo in conduzione non appena la tensione V_D diviene positiva. Questo equivale a supporre che la tensione di soglia Vγ e la caduta di tensione ai capi del diodo in conduzione siano nulle. Questa semplificazione non influisce sulla comprensione dei circuiti presentati. Nei risultati delle simulazioni presentati sarà invece evidente come il simulatore consideri la tensione di soglia e la caduta del diodo reale. Gli schemi di Cadlogix sono disponibili nel sito della rivista.

Raddrizzatore ad un semionda (clipper, limitatore)

Un circuito "limitatore" (clipper) è un circuito che lascia passare solo quella parte di segnale che si trova sopra o sotto un particolare valore di riferimento. Il valore di riferimento è costituito in tal caso dalla tensione di soglia del diodo.

Il circuito (figura 1), presentato nella precedente puntata, è costituito da un diodo e da una resistenza. Se alimentato con un segnale sinusoidale, il circuito lascia passare solo la parte di segnale di valore superiore alla tensione di soglia. Essendo questa assunta nulla per semplicità, ritroviamo in uscita solo le semionde positive. Nel caso di diodo reale, troveremmo in uscita solo la parte di segnale superiore a Vγ (figura 2).

Il valore medio della tensione di uscita vale

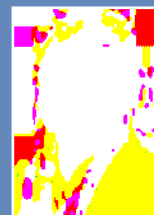
$$V_{um} = \frac{V_{uM}}{\pi}$$

dove V_{uM} è il valore massimo della sinusoide pulsante.

Domanda
Sapreste spiegare con chiarezza la differenza che c'è tra il valor medio di una forma d'onda ed il suo valore efficace?

Questo semplice circuito può essere usato come base per realizzare un caricabatterie, sebbene presenti l'inconveniente di fornire una tensione di uscita nulla durante la

Una panoramica sui diodi

di Riccardo Nicoletti
r.nicoletti@farelettronica.com

semionda negativa del segnale di ingresso.

Raddrizzatore di picco

Consideriamo il semplice circuito di figura 3, formato da un diodo e da un condensatore. Supponiamo che il condensatore sia inizialmente scarico e che il segnale di ingresso venga applicato, come in figura 3, a partire dall'istante t=0.

Quando V_i assume per la prima volta valori positivi il diodo conduce e la capacità comincia a caricarsi. Poiché supponiamo che non cada tensione ai capi del diodo, si ha $V_u=V_i$ (altrimenti $V_u=V_i - V_D$).
Il segnale di uscita "*insegue*" quello di ingresso. Dopo aver raggiunto il primo picco (V_M), il valore di V_i decresce mentre in uscita si ha $V_u=V_M$. Il diodo in questa situazione è interdetto e quindi non permette al condensatore di scaricarsi. Poiché la tensione in ingresso non diventerà mai maggiore di V_M, il diodo non condurrà più ed il condensatore rimarrà carico a V_M. La tensione di uscita rimane costantemente agganciata al valore V_M. L'unico caso in cui la tensione di uscita potrà variare sarà quello in cui Vi diverrà maggiore di V_M, ed in tal caso la tensione di uscita inseguirà il nuovo picco della tensione di ingresso (vedi Figura 5).

Figura 1 Circuito limitatore diodo-resistenza

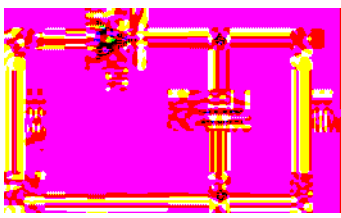



Figura 3 Raddrizzatore di picco

Questo circuito, detto "*raddrizzatore di picco*" converte una tensione sinusoidale in una continua di valore pari al *valore di picco* di quella sinusoidale.
Questo circuito trova applicazione nei radioricevitori, nei convertitori analogico-digitali e nei voltmetri elettronici.

Circuito rettificatore (diodo condensatore resistenza)

Per ovviare all'inconveniente del circuito di figura 1 (e successiva variante), si pone a valle del diodo un condensatore (generalmente elettrolitico) che abbia la possibilità di scaricarsi su un

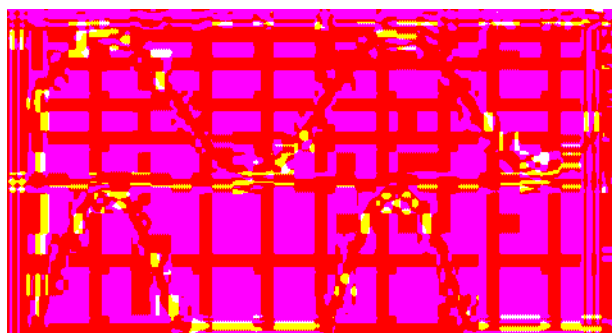



Figura 2 Simulazione del circuito limitatore

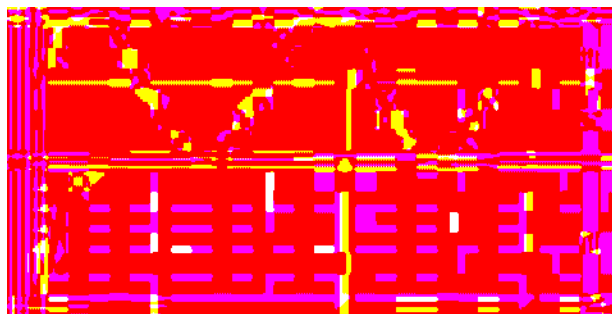



Figura 4 Simulazione del circuito raddrizzatore di picco con ingresso sinusoidale - si noti che il simulatore tiene conto della caduta ai capi del diodo: $V_u=V_M$ - 0.7V

carico R (figura 6). In tal caso il condensatore si carica quando il diodo è in conduzione, mentre si scarica su R quando il diodo è interdetto.
Questo consente al carico di ricevere tensione anche durante la semionda negativa che interdice il diodo.

In figura 7a si noti che la capacità C=1 µF si scarica piuttosto velocemente, rispetto al periodo della sinusoide di ingresso. La figura 7b riporta il risultato di una simulazione in cui il valore di capacità è maggiore. Sebbene la tensione di uscita non possa definirsi "continua" la situazione è migliorata rispetto al caso precedente. Per ottenere un risultato accettabile il condensatore elettrolitico utilizzato deve possedere un elevato valore di capacità, solitamente superiore ai 4700 µF. Esistono delle regole empiriche di grande utilità, riportate nei più comuni manuali, per calcolare il valore appropriato di capacità da inserire nel circuito raddrizzatore.

Un altro parametro importante per la scelta del condensatore è la sua tensione di lavoro. È buona norma scegliere un valore di tensione di lavoro superiore alla tensione massima della sinusoide raddrizzata. Ad esempio con V_M = 25V si può scegliere condensatori con tensione di lavoro 35V-50V.

Circuito clamper

Un circuito raddrizzatore di picco viene detto "*clamper*" (da "to clamp"=stringere, tenere in una morsa) quando l'uscita viene prelevata ai capi del diodo, anziché ai capi del condensatore (figura 8).

Nel primo quarto di periodo, quando il diodo conduce, la tensione di uscita è nulla. È nulla se consideriamo il diodo "ideale", mentre vale 0.6-0.7 V se non trascuriamo la caduta di tensione ai suoi capi.
Dopo il primo quarto di periodo il condensatore si è caricato e non può più scaricarsi in quanto il diodo è polarizzato inversamente:

$$V_i=V_c + V_D$$

Ma V_c è fissa al valore V_{iM}

Quindi

$$V_u=V_i - V_{iM}$$

Il segnale di uscita ha valor medio pari a $-V_{iM}$ e la forma d'onda è la stessa del segnale di ingresso.
Il circuito, detto spesso "*fissatore*" viene chiamato

Figura 5 *Circuito raddrizzatore di picco con ingresso variabile*

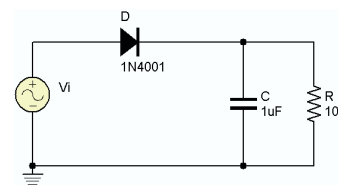



Figura 6 *Circuito rettificatore*

Figura 7a *Simulazione per il circuito rettificatore, con f= 50 Hz e C=1 µF*

Figura 7b *Simulazione per il circuito rettificatore, con f= 50 Hz e C=10 µF*

"clamper" poiché il picco positivo di V_i viene "costretto" al livello zero (figura 9).

Circuito raddrizzatore a doppia-semionda

Il *raddrizzatore a doppia semionda* (full wave rectifier) permette di raddrizzare entrambe le semionde del segnale sinusoidale presente in ingresso. Il valore medio dell'onda (positiva) che si ottiene in uscita è il doppio di quello ottenuto col raddrizzatore a singola semionda.

$$V_{um} = 2\,\frac{V_{uM}}{\pi}$$

I raddrizzatori a due semionde possono essere del tipo con trasformatore a presa centrale o con ponte di Graetz. Il raddrizzatore con trasformatore a presa centrale è riportato in figura 10.

Il segnale di ingresso V_i viene riportato dal trasformatore sul secondario, dove si hanno due segnali V_s aventi il riferimento di figura 10.

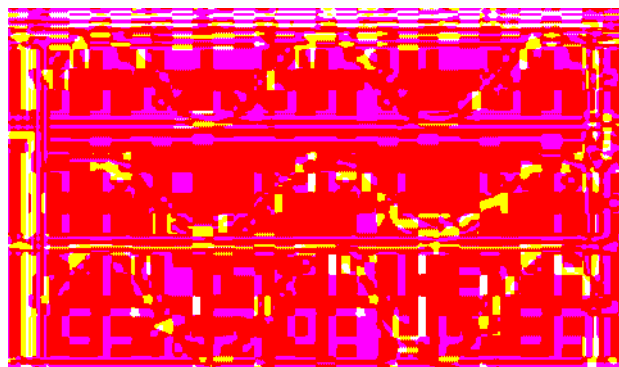



Figura 9 Simulazione del circuito clamper, con f= 50 Hz ed ampiezza 5V - la seconda curva è la tensione ai capi del diodo: si noti la caduta sul diodo che rende la forma d'onda leggermente superiore a 0V - la terza curva è la tensione ai capi del condensatore

Figura 10 Raddrizzatore con trasformatore a presa centrale

Quando il segnale V_i è positivo il segnale V_s riferito alla presa centrale (quello superiore) polarizza direttamente il diodo D1, mentre D2 è interdetto (figura 10a); quando V_i è negativo il segnale V_s superiore polarizza inversamente D1, mentre il segnale V_s inferiore, grazie alla presa centrale, polarizza direttamente D2 (figura 10b). Grazie a questa configurazione la corrente attraversa il carico sempre nello stesso verso e la tensione V_u è sempre positiva (figura 10c).

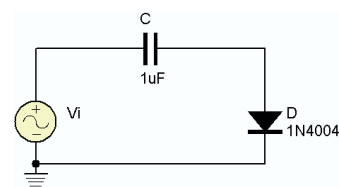



Figura 8 Circuito clamper

Figura 10a Durante la semionda positiva il diodo D1 conduce mentre D2 è interdetto

Figura 10b Durante la semionda negativa, per il riferimento imposto dalla presa centrale, il diodo D2 conduce, mentre D1 è interdetto

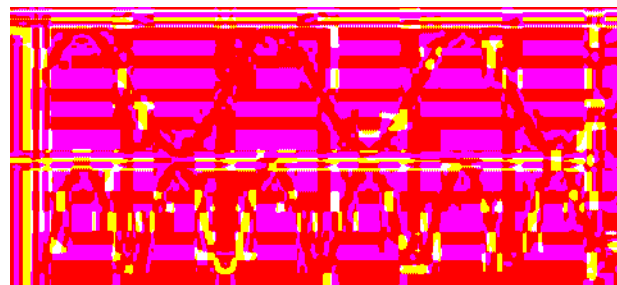



Figura 10c Risultato del simulatore Cadlogix per il raddrizzatore a presa centrale. Le semionde arrivano al carico con la stessa polarità, rendendo Vu positiva, con il riferimento di figura

Il diodo Schottky

Il diodo Schottky è formato da una giunzione del tutto particolare, costituita da un metallo (Alluminio) e da un semiconduttore (di tipo n). Questa struttura, denominata *giunzione "metallo-semiconduttore"* costituisce un elemento raddrizzante caratterizzato da un risposta molto veloce in fase di commutazione.

Un normale diodo a semiconduttore possiede una caduta di tensione in polarizzazione diretta che va da 0.7 V a 1.7 V, mentre il diodo Schottky fornisce una caduta che non supera 0.45 V. La sua tensione di soglia è intorno a 0.2 V ed il tempo di recupero inverso può raggiungere 1 ns.

I diodi Schottky vengono utilizzati quando sono richieste particolari prestazioni in commutazione, basse cadute di tensione e valori di soglia inferiori a quelli di un diodo standard. Si vedano, ad esempio, le considerazioni di *Romano Bernarducci* alle pagg. 51-53 del numero di Marzo.

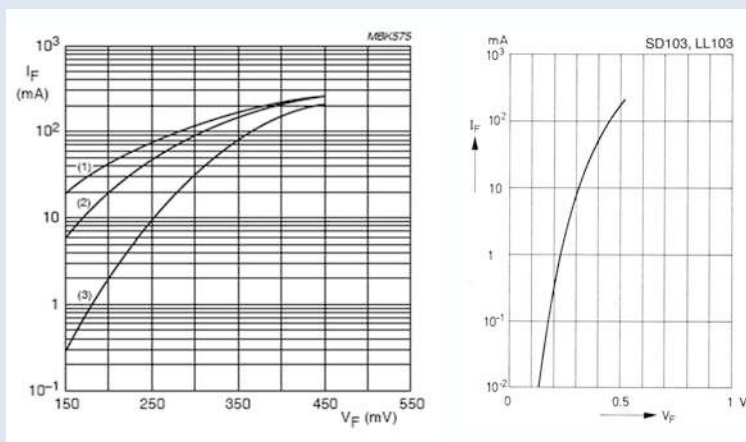



Figura 16 *Alcune caratteristiche di diodi Schottky - un datasheet è disponibile nel sito della rivista*

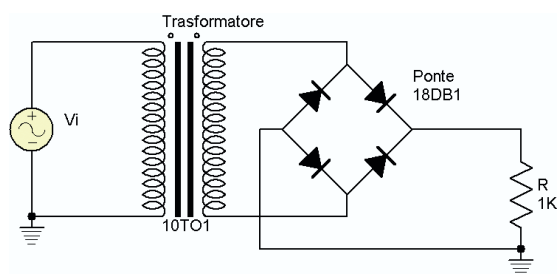



Figura 11 *Circuito raddrizzatore a ponte di Graetz con trasformatore*

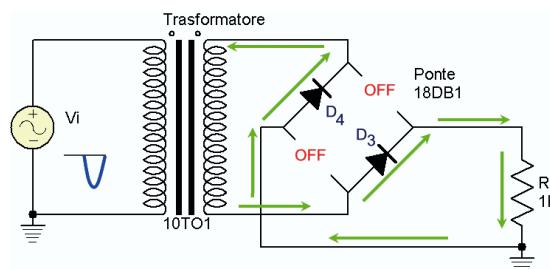



Figura 13 *La semionda negativa di Vi rende positivo il morsetto inferiore del secondario del trasformatore, polarizzando direttamente il diodo D3.*

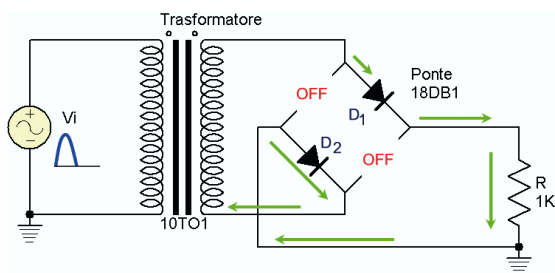



Figura 12 *La semionda positiva è riportata sul secondario del trasformatore in modo da polarizzare direttamente il diodo D1*

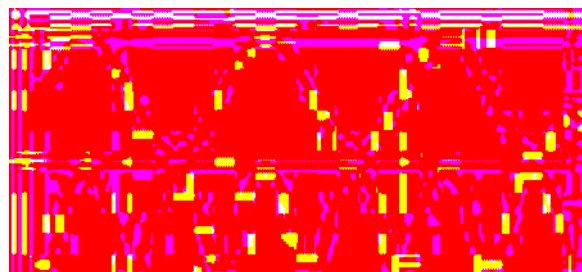



Figura 14 *Simulazione del raddrizzatore a ponte di Graetz. Si noti che in questo caso la tensione di uscita risente due volte della caduta ai capi del diodo, quindi* $V_{UM}=V_{IM}-1.4V$

Il raddrizzatore più utilizzato è tuttavia quello a ponte di *Graetz*, utilizzato nello schema in figura 11 a valle di un trasformatore con rapporto spire 10:1.

Il raddrizzatore è formato da quattro diodi, disposti a ponte. Durante la prima semionda, positiva, conducono i diodi D1 e D2, mentre gli altri due sono interdetti (figura 12); durante la semionda successiva conducono D3 e D4, mentre D1 e D2 sono interdetti (figura 13). In questa configurazione la corrente (indicata in verde) percorre il carico sempre nello stesso verso e quindi la tensione ai suoi capi è sempre positiva (figura 14).

Il raddrizzatore a doppia semionda è disponibile in commercio sotto forma di dispositivo "pronto all'uso"; quattro diodi collegati a ponte sono incapsulati in una resina epossidica ed accessibili dai soli terminali esterni (figura 15). A due terminali si applica la sinusoide e dagli altri due si preleva la sinusoide raddrizzata. Questo componente viene posto a valle del trasformatore e costituisce il primo elemento per la realizzazione di un buon alimentatore. Esso viene scelto in base alla tensione massima presente in ingresso ed alla corrente da erogare.

Negli stabilizzatori di tensione di buona qualità, il raddrizzatore a ponte viene fatto seguire da un condensatore elettrolitico di grande valore (es. 4700 μF-2200 mF), ai capi del quale si preleva una tensione con un piccolo "*ripple*". Tale tensione entra in uno stabilizzatore integrato (ad esempio un circuito della serie 78XX) che provvede a fornire in uscita una tensione perfettamente costante.

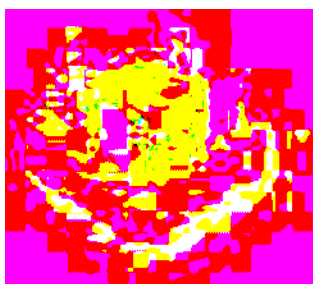



Figura 15 *Un raddrizzatore a ponte di Graetz di tipo commerciale*

Figura 17 *Alcuni LED di tipo commerciale*

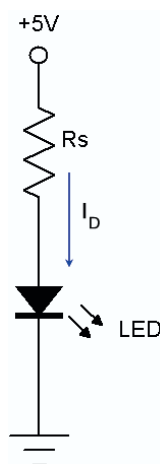



Figura 18 *Un semplice circuito per accendere un diodo LED*

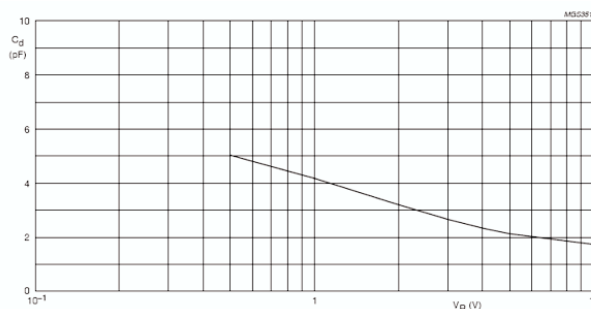



Figura 19 *Caratteristiche del diodo varicap BB141*

IL DIODO LED

Il diodo emettitore di luce, o LED (acronimo di Light Emitting Diode), è un particolare diodo capace di emettere luce quando attraversato da corrente. La sua giunzione può essere formata da diversi materiali, a seconda dei quali cambia il colore della luce emessa. Nel caso di giunzione con Arseniuro di Gallio si ha luce rossa, con Fosfuro di Gallio la luce diviene gialla o verde (figura 17).
Tralasciamo la spiegazione del fenomeno che porta all'emissione di luce, rimandando i Lettori più curiosi a testi specializzati.

Il LED è un diodo a tutti gli effetti, e possiede dunque un anodo e un catodo. Affinché la corrente che lo attraversa provochi la giusta luminosità, lo si pone sempre in serie ad un resistore di limitazione (figura 18) .

Il resistore di limitazione R_s si dimensiona in base alla corrente che deve attraversare il LED, di valore variabile da 20 a 50 mA a seconda del tipo di luce. La caduta ai capi del LED può raggiungere 2-2.7V.
Un esempio di datasheet della Roithner Laser-technik è disponibile nel sito della Rivista.

Figura 20 *Caratteristriche della famiglia di diodi Zener BZX55 (datasheet ITT)*

IL DIODO VARICAP

Abbiamo visto come una giunzione pn diventi sede di una capacità, detta di *transizione o di carica spaziale*, dovuta alla carica spaziale degli ioni donatori ed accettori non neutralizzati su entrambi i lati della giunzione. Questa capacità non è costante, ma varia con la tensione applicata al diodo ed è importante nel caso di polarizzazione inversa.

I diodi costruiti per sfruttare la variazione di questa capacità si chiamano *varactors o varicaps*. Un diodo varactor non è altro che un dispositivo a semiconduttore utilizzato come elemento a reattanza variabile.

Le applicazioni maggiori dei diodi varicap si hanno nel campo delle microonde, in generatori di armoniche, modulatori, circuiti di controllo automa-

Figura 21 *Un diodo Zener commerciale, con uno dei simboli grafici più utilizzati*

Rs
Vi
Dz
BZX55C6V8
RL

Figura 22 *Stabilizzatore con diodo zener*

tico della frequenza e in circuiti risonanti.
La capacità dei diodi varicap può variare da pochi picofarad a qualche centinaio di picofarad, a seconda delle applicazioni.
In figura 19 è riportata la caratteristica capacità-tensione per il diodo BB141. Il data-sheet completo è disponibile nel sito www.farelettronica.com.

IL DIODO ZENER

Il diodo Zener (dal nome dello scienziato Clarence Zener, che studiò l'effetto su cui si basa il dispositivo) è un particolare tipo di diodo, costruito appositamente per funzionare nella zona di polarizzazione inversa. In tale zona, dove la caratteristica diviene più rettilinea, a grandi escursioni di corrente corrispondono piccole variazioni di tensione (vedi figura 20). Entro certi limiti la tensione rimane pari al valore costante V_z.

Si noti nelle caratteristiche di figura 20, il valore di corrente denominato "test current I_z", corrispondente al *ginocchio* della caratteristica. Il funzionamento stabilizzatore del diodo si ha per valori di corrente superiori ad I_z (altre volte denominata I_k da "*knee*"=ginocchio).
Il diodo Zener si presta dunque per realizzare un semplice stabilizzatore di tensione, o meglio per creare una tensione di riferimento.
Ci preme evidenziare che lo stabilizzatore con diodo Zener è solo un primo esempio di circuito stabilizzatore (oltremodo economico!), in quanto non offre l'accuratezza e l'affidabilità di un circuito stabilizzatore integrato.
La tensione fornita dal diodo Zener è impressa sul suo involucro.
Consideriamo il circuito di figura 22. Notate prima di tutto che il diodo è polarizzato inversamente: il suo catodo è rivolto verso la tensione positiva.

La tensione di ingresso si chiama V_i, la resistenza di carico è R_L mentre R_s è la resistenza che mettiamo nel circuito per limitare la corrente I_z che attraversa lo Zener.
Il valore che verrà stabilizzato sul carico è V_z, di valore inferiore alla tensione di ingresso del circuito. Se omettiamo il resistore R_s sentiremo un lieve odore di bruciato dopo pochi secondi...
Se V_z è la tensione sul catodo del diodo, la corrente che attraversa R_s è

$$I_s = \frac{V_i - V_z}{R_s}$$

e questa corrente comprende le due correnti I_z ed I_L, quindi

$$R_s = \frac{V_i - V_z}{I_z + I_L}$$

il valore I_z è quel valore di corrente che occorre far scorrere nel diodo affinché la tensione ai suoi capi V_z sia stabilizzata.
Vediamo un esempio. Supponiamo di avere una tensione di 12V e di volerla stabilizzare a 9V per alimentare un circuito che assorbe 10 mA. Prima di tutto notiamo che non c'è uno Zener da 9V, ma dobbiamo accontentarci di uno da 9.1V. Facciamo scorrere nel diodo una corrente inferiore a quella massima, per esempio 12 mA.

$$R_s = \frac{V_i - V_z}{I_z + I_L} = \frac{12 - 9{,}1}{12 + 10} \times 1000 = 131{,}8\Omega$$

non essendo disponibile in commercio un valore simile, scegliamo R_s=120Ω. Con questo valore la corrente che scorre attraverso R_s vale

$$I_s = \frac{V_i - V_z}{R_s} = \frac{12 - 9{,}1}{120} = 24mA$$

e quindi quella che attraversa lo Zener è

$$I_z = I_s - I_L = 24 - 10 \text{ mA} = 14 \text{ mA}.$$

Come mai lo Zener "stabilizza"?

Eventuali variazioni della tensione di ingresso, che si ripercuotono sul circuito come variazio-

ni di corrente su R_s, vengono assorbite dal diodo Zener, dato che sul carico scorre una corrente costante pari a V_z/R_L. Il dimensionamento di un semplice stabilizzatore condotto in questo modo non porta a forti attitudini a "stabilizzare" la tensione di uscita: si consideri infatti che se variasse il carico occorrerebbe ridimensionare ogni volta la resistenza R_s.
Inoltre questi stabilizzatori sono utili finché la corrente sul carico si aggira intorno a poche decine di milliampere. Un altro svantaggio è che il carico non può mai essere scollegato; se il progetto non fosse ben fatto, in tal caso, anche la corrente I_L attraverserebbe il diodo e lo danneggerebbe.
I diodi Zener possiedono inoltre una certa tolleranza sul valore di tensione stabilizzato. Questo significa che una tensione da 5.1 V può essere stabilizzata su 4.9 V o su 5.2 V.

Osservando la tabella di figura 23, si notano due cose importanti. La prima è che il valore della corrente di ginocchio I_{ztest} è pari a 5mA per tutti i diodi della famiglia.
La seconda è che la tensione di lavoro V_z (misurata in corrispondenza di I_{ztest}) possiede un'escursione di valori con tolleranza di ±5%. Questo ci fa capire che lo Zener non è un componente molto accurato. In ultima analisi vediamo anche che non sono disponibili tutti i valori di tensione V_z, ma solo alcuni.

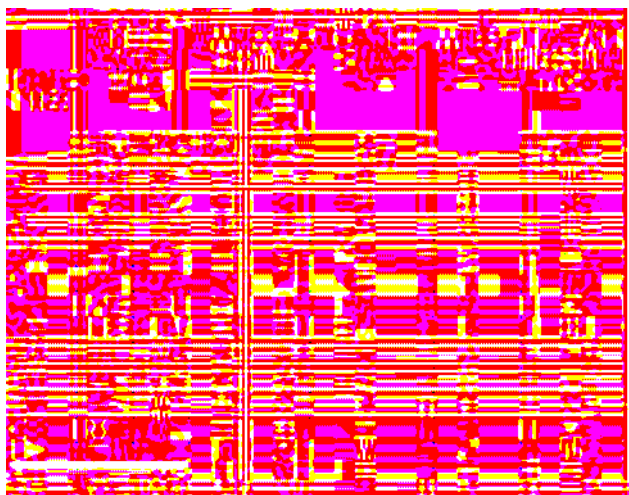



Figura 23 *Alcuni parametri dei diodi Zener della serie BZX55 (Philips Semiconductors)*

Supponiamo di avere a disposizione una tensione da 9V e di volerla stabilizzare a 6.8V, su un carico che assorbe 10 mA. Scegliamo per il progetto un diodo Zener BZX55 C6V8 (figura 24) da 500 mW (il suo data-sheet lo trovate nel sito della Rivista). La corrente I_{ztest} di ginocchio è data pari a 5 mA, mentre la corrente massima sopportabile è I_{zM}= 58 mA. Le due regole da rispettare nel dimensionamento dello Zener sono: corrente I_z minima superiore a quella di ginocchio ($I_z > I_{zK}$) e corrente I_{zMAX} inferiore a quella massima sopportabile ($I_z < I_{zM}$).

Facciamo scorrere nel diodo una corrente inferiore a quella massima, ad esempio 25 mA, valore che si trova circa a metà tra il valore minimo ed il valore massimo consentito.

$$R_S = \frac{V_i - V_z}{I_z + I_L} = \frac{9 - 6{,}8}{25 + 10} \times 1000 = 62{,}86\Omega$$

prendiamo il valore resistivo commerciale più vicino a 62.86 cioè 68 ohm.

Con questo valore la corrente che attraversa R_s vale

$$I_s = \frac{V_i - V_z}{R_S} = \frac{9 - 6{,}8}{68} = 32mA$$

e poiché V_z rimane costante – entro una tolleranza sufficientemente stretta – questa corrente è quella che attraverserà R_s in ogni situazione.
Il diodo Zener è attraversato da una corrente $I_z = I_s - I_L = 32 - 10 = 22$ mA cioè un valore accettabile nel range di correnti del diodo.
Se scolleghiamo il carico (o se il carico diventa improvvisamente un circuito aperto) cosa accade al diodo?

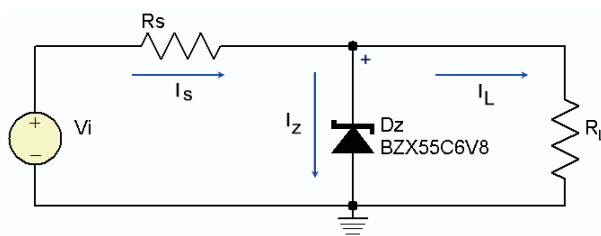



Figura 24 *Il circuito stabilizzatore con diodo Zener*

La corrente che passa in R_s attraversa tutta il diodo, quindi $I_z = 32$ mA perfettamente al di sotto del valore massimo consentito.

Supponiamo che la corrente nel carico aumenti del 50%, cioè $I_L = 20$ mA.
La corrente che attraverserà il diodo in tale situazione diverrà $I_z = I_s - I_L = 32 - 20 = 12$ mA. Valore che consente al diodo di funzionare correttamente.

L'intervallo di tensioni di ingresso che il diodo riesce a stabilizzare alla tensione di Zener è limitato dalle due condizioni:

$$I_z > I_{zK} \Rightarrow V_i > (I_{zK} + I_L) R_s + V_z$$

$$I_z < I_{zM} \Rightarrow V_i < (I_{zM} + I_L) R_s + V_z$$

e nel nostro caso

$$V_i > (I_{zK} + I_L) R_s + V_z = (5mA + 10mA)\ 68 + 6.8 = 7.82\ V$$

$$V_i < (I_{zM} + I_L) R_s + V_z = (58mA + 10mA)\ 68 + 6.8 = 11.42\ V$$

La figura 26 riporta la simulazione del circuito nel caso in cui la tensione di ingresso sia una tensione continua da 9V con sovrapposta una lieve ondulazione a 20 Hz.
Si noti come le variazioni di tensione e corrente sul carico risentano dell'effetto stabilizzatore dello Zener.
Ulteriori approfondimenti possono essere fatti per il circuito contenente il diodo Zener. È possibile apportare delle migliorie ed ottimizzare la stabilizzazione nei confronti di variazioni della tensione di ingresso o del carico. Rimandiamo i Lettori più curiosi a testi specializzati.

PROSSIMA PUNTATA

Nella prossima puntata ci occuperemo del "principe" dei dispositivi a semiconduttore, ovvero il transistore. Cercheremo di darne una breve descrizione e di comprenderne il funzionamento. Alcuni semplici circuiti ci aiuteranno nell'intento.

BIBLIOGRAFIA

1. J. Millman, C. C. Halkias – "**Microelettronica**", Ed. Boringhieri, 1978, Torino
2. M. Gasparini, D. Mirri – "**Dispositivi e circuiti elettronici**", Vol. I, III ed., Ed. Calderini, Bologna, 1983
3. A. Montanari – "**Tecnologia delle costruzioni elettroniche**", Vol. II, Edizioni Cupido, 1983
4. E. Borchi, R. Nicoletti – "**Circuiti elettrici lineari**", Vol. I, Ed. Esculapio, 1999, Bologna

Per i datasheet dei diodi:

5. ITT Semiconductors – "**Diodes, Zener Diodes, Rectifiers**", 1985
6. Philips Semiconductor *www.semiconductors.philips.com*

Per i datasheet dei diodi LED:

7. Roithner Lasertechnik, Austria *www.roithner-laser.com*

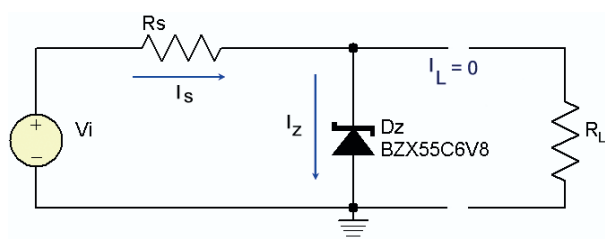



Figura 25 *Situazione in cui il carico viene scollegato*

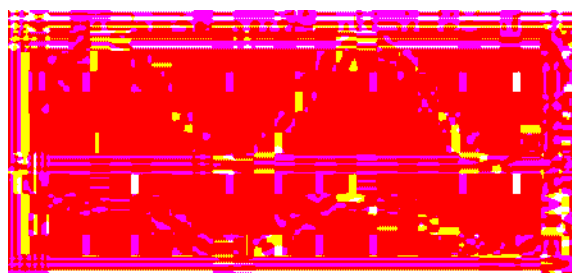



Figura 26 *Simulazione del circuito stabilizzatore con diodo Zener*

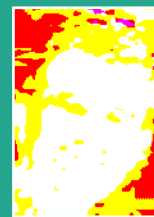
Smartcard

Quanti di voi hanno dovuto cambiare, per un motivo o per un altro, la propria SIM? Avevate perso la precedente, avete cambiato operatore telefonico, avete ricevuto una nuova SIM più potente e capiente dello stesso gestore.
In tutti questi casi, vi sarete imbattuti in un noiosissimo problema: trasferire la vostra rubrica sulla nuova scheda.

COMINCIAMO UNA NUOVA PARTITA A CARTE

Cosa ci siamo detti negli articoli precedenti

Come previsto nell'ultimo articolo del tutorial sulle smartcard, pubblicato in sette puntate su Fare Elettronica a partire dal n. 227 di Maggio 2004 al n. 234 del Dicembre 2004, si è presentata l'occasione per cominciare una nuova partita a carte, o meglio… a smartcard. In questo articolo, verranno utilizzati numerosi concetti già sviluppati nel corso delle precedenti puntate del tutorial.
Qui farò solo un velocissimo riassunto.
Come i miei lettori ormai sanno molto bene, una SIM non è altro che una smartcard asincrona a microprocessore utilizzata, tra le altre cose, per conservare alcuni dati utili all'utente (per esempio, la rubrica telefonica e gli SMS) e altri dati utili durante l'accesso alla rete (per esempio, il gestore telefonico e il numero seriale della SIM, insieme alla chiave segreta utilizzata per l'autenticazione).
Le SIM, come molte altre smartcard che seguono lo standard ISO7816-4, presentano un'interfaccia con l'esterno che mostra una organizzazione razionale dei dati sotto forma di un file-system, formato da EF (Elementary File) e DF (Dedicated File). I primi sono paragonabili ai file presenti in un normale hard-disk di un PC, mentre i secondi possono essere accomunati alle cartelle o directory. Tutti questi file sono organizzati ad albero, così come nella maggior parte dei sistemi operativi che abitualmente utilizziamo sui nostri computer, a partire da una cartella principale denominata MF (Master File). Così come nel buon vecchio DOS, oltre che nei molti sistemi operativi che mettono a disposizione shell di comandi, è possibile "navigare" all'interno di questo file system mediante appositi comandi da inviare alla SIM. Questi e gli altri comandi sono definiti in seno agli innumerevoli standard GSM emanati dall'ETSI (a tal proposito, si veda la bibliografia a fine articolo). Per esempio, il comando definito SELECT FILE permette di cambiare la "cartella corrente", indicandone l'identificativo di due byte, in modo simile al comando CD del DOS. Sempre con lo stesso comando, è possibile selezionare un EF per poi leggerlo e/o scriverlo.
Gli standard GSM definiscono alcuni EF e DF, il loro significato, la loro posizione nell'albero. Per esempio, l'EF denominato ADN ed indicato come EF_{ADN} è un file che si trova all'interno del $DF_{TELECOM}$ che, a sua volta, è contenuto nell'MF.
Questo file è quello che contiene la rubrica all'interno della nostra SIM.
Si tratta di un file basato su record, in cui ogni record rappresenta un nome in rubrica. Il numero e la dimensione di un record possono cambia-

di Giuseppe Modugno
gppe.modugno@libero.it

SIMCopier

re a seconda della capacità di memorizzazione della SIM fornita dal nostro operatore: più recente è la SIM, più numeri avremo a disposizione in rubrica e più lunghi saranno i nomi memorizzabili. La composizione di un singolo record dell'EF_{ADN} è mostrata in figura 1.

Il primo campo, la cui dimensione dipende dalle caratteristiche della SIM, è definito *alpha-identifier* ed è utilizzato per la memorizzazione del nome associato al numero telefonico. Gli altri campi del record, di lunghezza complessiva pari a 14 byte, conservano altre importanti informazioni associate ad un numero telefonico, tra cui il numero stesso. A chi volesse maggiori dettagli su questo e sugli altri file presenti in una SIM, consiglio di leggere la parte VI e VII del tutorial o di leggere gli standard relativi. La dimensione totale del record, da cui si può ricavare la lunghezza massima dei nomi, sottraendo 14, e il numero dei record possono essere ricavati dalla risposta al comando GET RESPONSE subito dopo aver selezionato il file EF_{ADN}.
Così come un telefonino, anche un lettore di smartcard può accedere al file-system contenuto nella SIM, selezionare i file e leggere/scrivere il contenuto. Per un file basato su record, esistono due comandi utilizzati per la lettura e la scrittura: READ RECORD e UPDATE RECORD. In entrambi i casi, l'operazione avviene con successo solo se è soddisfatta la condizione d'accesso, cioè se è già stato inserito il CHV1 (PIN), se questi è abilitato. In caso contrario, prima di poter leggere e/o scrivere i record in rubrica, sarà necessario procedere all'inserimento del PIN, mediante il comando VERIFY CHV.

Figura 1 *Rappresentazione di un record del file EF_{ADN}*

INTRODUZIONE

L'Idea

Poco tempo fa ho accettato una proposta del mio operatore di telefonia mobile che mi invitava a sostituire la mia vecchia SIM GSM (100 numeri in rubrica da 12 caratteri e 10 SMS) con un'altra moderna e più capiente (250 numeri in rubrica da 18 caratteri e 50 SMS). Ovviamente i vantaggi della nuova SIM non si limitano alla capacità di memorizzazione ma, ai fini di questo articolo, è quello che ci interessa maggiormente. Deciso ad effettuare questa conversione di scheda, mi reco in un punto vendita autorizzato dove mi forniscono la nuova e fiammante SIM che sarebbe stata abilitata dopo qualche ora. La simpatica addetta mi chiede se volessi fare il trasferimento automatico della rubrica sulla nuova SIM, cosa che ha subito stimolato la mia curiosità. Alla mia risposta affermativa, la commessa inserisce in un piccolo apparecchio da tavolo, utilizzando gli appositi adattatori per le carte SIM, la mia vecchia SIM e la nuova, digita, tramite apposita tastiera, i due PIN e avvia l'operazione. Dopo qualche minuto, tutta sorridente mi riconsegna entrambe le SIM. Inutile dire che la mia rubrica era stata effettivamente copiata sulla nuova SIM, in modo indolore. Naturalmente, l'idea di progettare un apparato simile che mi permettesse di copiare la rubrica delle SIM m'è venuta immediatamente, portandomi alla realizzazione del SIMCopier e, ovviamente, a questo articolo.

Descrizione

Cos'è dunque questo SIMCopier? Dal nome stesso, si intuisce subito che si tratta di un circuito che effettua la copia di una SIM.
Tecnicamente quello che viene copiato è il solo file EF_{ADN} che contiene la rubrica telefonica. Ciò può essere utile in numerosissime situazioni. Per esempio, come nel mio caso, quando si vuole utilizzare una SIM più capiente dello stesso operatore. Ma questa è solo una delle possibili situazioni in cui il SIMCopier può rivelarsi utile. Esso può essere un toccasana quando perdete la vostra SIM: avendo fatto in precedenza la copia della rubrica mediante SIMCopier, potete tranquillamente recuperarla dalla SIM di backup.
Ancora, se volete cambiare operatore potete trasferire la rubrica dalla vecchia SIM del vostro precedente gestore alla SIM del nuovo operatore telefonico. Il SIMCopier è, infatti, compatibile con tutte le SIM, indipendentemente dall'operatore e può effettuare la copia della rubrica tra due SIM di diversi gestori.
La scheda che dovremo progettare sarà quindi formata almeno da due connettori per smartcard, un tasto e qualche led di indicazione. Un connettore ospiterà la SIM sorgente, quella che contiene la rubrica da copiare, l'altro verrà utilizzato per la SIM destinazione, dove la rubrica verrà copiata.
Uno dei problemi che ho dovuto affrontare è stato quello relativo all'inserimento del PIN. Come tutti voi sapete, per poter accedere alla rubrica della vostra SIM, sia in lettura che in scrittura, è necessario conoscere ed inserire il

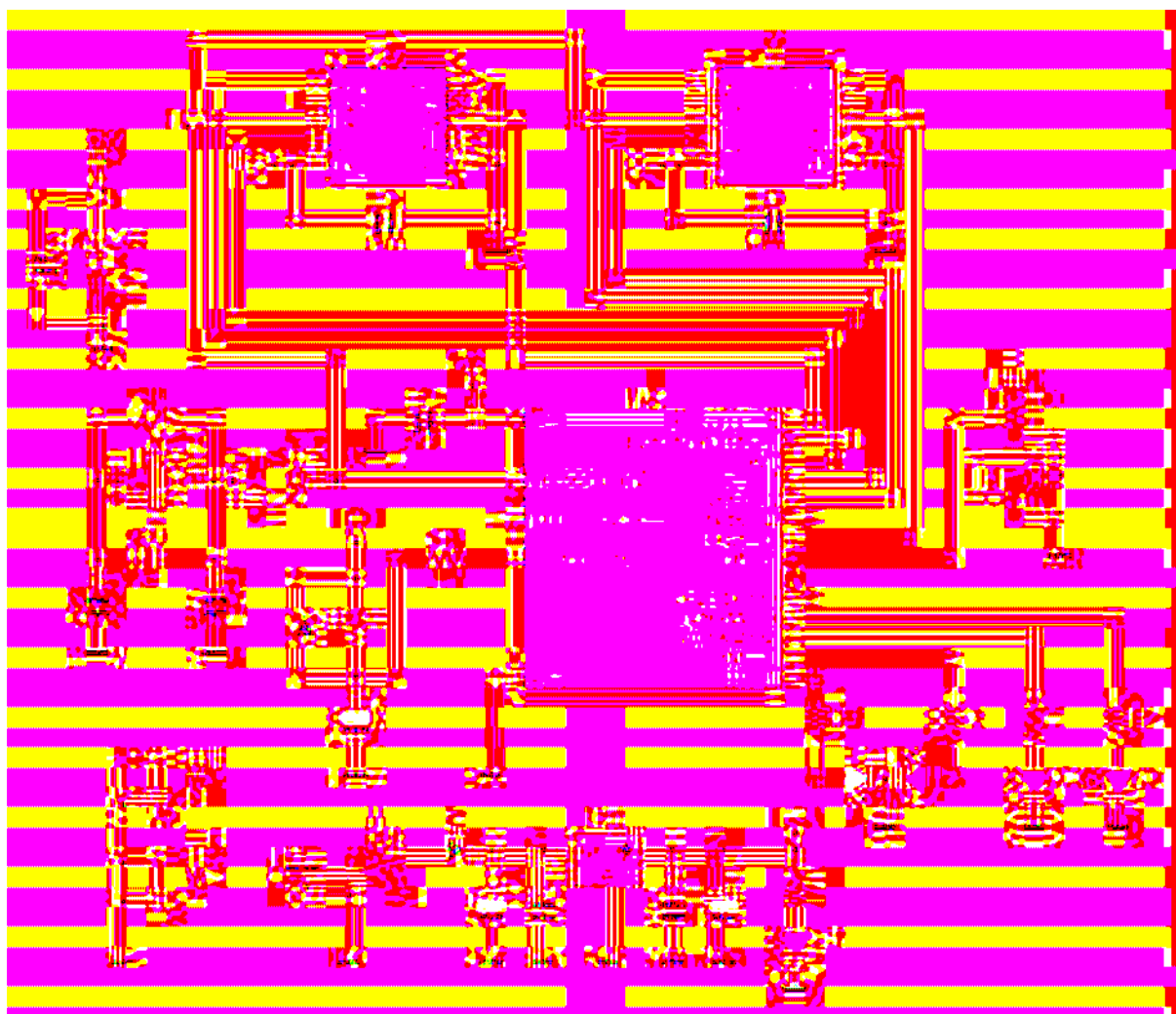



Figura 2 *Schema elettrico*

PIN corretto. Questo avrebbe complicato notevolmente il prototipo, poiché avrei dovuto prevedere almeno un display ed un tastierino numerico, mentre la mia idea era quella di un copiatore di SIM veramente compatto e semplice. Così ho sfruttato la possibilità di disabilitare il PIN, per esempio tramite un qualsiasi telefonino (dall'apposito menu sulle impostazioni della sicurezza) oppure tramite un lettore di smartcard (come l'UniReader proposto nella parte III del tutorial), inviando l'apposito comando DISABLE CHV.
Quindi, per poter effettuare la copia della rubrica con il nostro SIMCopier, sarà necessario prima di tutto disabilitare, su entrambe le schede, il PIN.
Fatto questo, sarà sufficiente inserire le due SIM all'interno dei rispettivi connettori, *rispettando la posizione della SIM sorgente (SOURCE)* e *destinazione (DESTINATION)*, alimentare il SIMCopier e premere il tasto di avvio copia (COPY).

SCHEMA ELETTRICO

Lo schema elettrico del SIMCopier è mostrato in figura 2. La sezione di alimentazione (in basso nello schema elettrico) è formata dal solito regolatore lineare di tensione 7805 (IC3), che permette di avere, in uscita, una tensione continua stabilizzata a 5V, che servirà per alimentare tutti gli integrati e le smartcard del circuito. La tensione d'ingresso dovrà essere applicata al comunissimo connettore J1, molto spesso già presente negli alimentatori a muro.
Questa tensione non dovrà essere necessariamente stabilizzata, poiché sarà il regolatore 7805 a mantenerla costante. Si può quindi usare un semplice trasformatore seguito da un

ponte di diodi. La tensione minima necessaria per far funzionare correttamente il 7805 è di circa 8V, ma può essere aumentata anche fino a 24V. Il polo positivo del connettore J1 è quello centrale. Il diodo D2 è utilizzato per bloccare la corrente nel caso in cui la tensione di ingresso venisse applicata con polarità opposta. Il led verde DL1 è collegato direttamente alla 5V, quindi si accende non appena viene data alimentazione al circuito. Esso può essere utilizzato per controllare il corretto funzionamento della sezione di alimentazione del SIMCopier.
Come si può osservare, il cuore di tutto il circuito è rappresentato dall'integrato PIC16F628 (IC1), cioè un diffusissimo microcontrollore a 8 bit della Microchip con memoria Flash da 2048 istruzioni per il codice, 128 byte di EEPROM e 224 byte di RAM. È lo stesso microcontrollore utilizzato nel lettore di smartcard UniReader, anche se in modo diverso. In questo caso, infatti, il clock necessario al micro per la corretta esecuzione del programma proviene da un oscillatore esterno formato dall'integrato 74HC00 (IC2), dal quarzo a 3.579MHz (Q1), dai due condensatori C1 e C2 e dalla resistenza R1. La strana frequenza utilizzata, tipicamente presente nei lettori di smartcard, permette una comunicazione seriale con la carta alla velocità standard di 9600bps (per una esauriente spiegazione si rimanda alla parte III del tutorial). Per questo motivo, il clock generato dall'oscillatore pilota non solo il microcontrollore, ma anche le due smartcard.
Naturalmente in fase di programmazione del PIC, sarà necessario impostare la parola di configurazione sulla modalità EC (External Clock). In questa modalità, verrà disabilitato l'oscillatore interno del microcontrollore ed utilizzato il clock presente al pin CLKIN (pin 16). Gli ingressi delle porte NAND non utilizzate del 74HC00 sono collegate a massa per evitare di lasciarle in condizione di alta impedenza.
La sezione formata dal condensatore C10, dalle resistenze R2 e R8 e dal diodo D1, forma il circuito di reset del microcontrollore. Il PIC necessita di un livello alto sul pin MCLR (Master

Numero	Nome	Direzione	Descrizione
C1	VCC	I	Alimentazione della smartcard, normalmente a 5V, anche se sono previste alimentazioni a tensioni inferiori (3V o meno)
C2	RST	I	Linea di reset utilizzata dal dispositivo di lettura per inizializzare la scheda, prima di un qualsiasi trasferimento di dati
C3	CLK	I	Linea utilizzata per fornire il clock ai circuiti sequenziali integrati nella smartcard (carte asincrone), oppure per lo scambio dei dati (carte sincrone)
C4	RFU		Riservata per usi futuri
C5	GND	I	Massa
C6	VPP	I	Tensione di programmazione, eventualmente utilizzata per la programmazione (cancellazione/scrittura) della memoria non volatile integrata nella smartcard (EEPROM, Flash)
C7	I/O	I/O	Linea bidirezionale utilizzata per lo scambio dei dati tra smartcard e dispositivo di lettura
C8	RFU		Riservata per usi futuri

Tabella 1 Gli 8 contatti di una smartcard ISO7816 (la direzione è relativa alla smartcard)

Clear) per poter iniziare l'esecuzone del programma memorizzato in Flash. All'accensione del circuito, il livello di tensione su questo pin sale lentamente seguendo la legge di carica del condensatore C10, limitata solo dalla resistenza R2. Questo permette di ritardare l'esecuzione del programma da parte del PIC in modo che il clock generato dall'oscillatore si stabilizzi. Il diodo D1, invece, forma un "canale veloce" di scarica del condensatore C10, quando viene tolta l'alimentazione al circuito. In questo modo, il microcontrollore viene immediatamente resettato per riprendere l'esecuzione non appena l'alimentazione ritorna a 5V.
Il tasto T1, collegato direttamente ad un pin del micro, viene utilizzato per iniziare la copia della rubrica. Quello utilizzato nel nostro prototipo è un tasto momentaneo che sposta, premendolo, due deviatori. Nel nostro circuito, viene utilizzato un solo deviatore, collegato al PIC, in modo che il pin relativo venga messo a massa alla pressione del tasto.
Quando il tasto non è premuto, il pin viene portato a livello alto grazie alla resistenza di pull-up interna al microcontrollore. Non è un caso, infatti, che ho collegato il tasto al pin RB0, in quanto per questo, e per tutti gli altri pin della porta B, possono essere abilitati dei pull-up via software (in particolare, mettendo a 0 il bit 7 del registro OPTION). È bene ricordare, però, che è possibile abilitare o disabilitare i pull-up su tutti i pin della porta B contemporaneamente (in altre parole, non è possibile, per esempio, abilitare un pull-up su RB0 e disabilitarlo su RB4).
I led DL2 e DL3 sono collegati direttamente ad un pin del PIC, tramite le resistenze di limitazione della corrente R4 e R5, e sono utilizzati come

segnalazione di presenza SIM. L'accensione del led DL2 indica la presenza di una SIM nello slot destinazione; l'accensione di DL3 indica la presenza di una SIM nello slot sorgente.
In questo modo, prima di iniziare la copia premendo l'apposito tasto, è possibile accertarsi che le due SIM siano correttamente inserite nei rispettivi connettori. Ciò è molto importante soprattutto se utilizziamo, come molto probabilmente accadrà, degli adattatori per SIM che, se di scarsa qualità, potrebbero non effettuare un buon contatto tra i pin delle carte e i pin del connettore. I led potranno essere accesi via software semplicemente mettendo ad 1 l'uscita della porta relativa (RA2 per DL2 e RA3 per DL3). Ricordo che tale collegamento può essere fatto poiché il PIC16F628, come molti altri PIC, riesce a fornire delle correnti di source (in uscita dal pin) anche di

Carta inserita?
NO
Spegni LED
SI
Reset
Lettura ATR
Controllo della Sintassi dell'ATR
ATR valido?
NO
Spegni LED
SI
Accendi LED

Figura 3 *Diagramma di flusso del firmware relativo alla rilevazione della presenza di una SIM*

25mA, corrente sufficiente per l'accensione del led. Il componente DL4 è un bi-led, formato cioè da un led verde ed un led rosso connessi a catodo comune. È possibile accenderli singolarmente oppure contemporaneamente producendo, in quest'ultimo caso, un colore simile al giallo. Entrambi i led sono collegati, così come per DL2 e DL3, ad un pin del microcontrollore e sono usati per indicare lo stato di avanzamento della copia.
Alla pressione del tasto e durante la copia, vengono accesi entrambi i led (portando a livello alto il pin RA0 e RA1). Se la copia viene terminata con successo, viene acceso solo il led verde; se, invece, viene riscontrato qualche errore (per esempio, una delle SIM ha il PIN abilitato), viene acceso solo il led rosso.
L'ultima sezione del circuito riguarda l'interfacciamento tra i due connettori per smartcard ed il microcontrollore. Come per l'UniReader, ho utilizzato uno slot ISO7816 dotato di un interruttore normalmente chiuso che si apre all'inserimento di una smartcard. Quindi, ogni connettore ha 10 pin: 8 rappresentano i contatti della smartcard (C1-C8) e due rappresentano quelli dell'interruttore (SW1 e SW2). Così come per il tasto T1, gli interruttori dei due slot sono collegati direttamente ad un pin del micro. Normalmente tali pin sono a livello basso, poiché connessi direttamente a massa tramite l'interruttore; inserendo una smartcard, l'interruttore si apre ed il pin si porta a livello alto grazie ai soliti pull-up interni. Infatti, i pin utilizzati, RB4 e RB5, appartengono alla porta B. In questo modo, il microcontrollore può accorgersi dell'inserimento o disinserimento di una smartcard. Gli 8 contatti di una generica smartcard sono elencati in tabella 1.
Di questi, solo 6 sono specificati dagli standard ISO, mentre gli altri due contatti (C4 e C8) sono riservati per usi futuri. Inoltre, nelle SIM anche il contatto Vpp (C6) non viene utilizzato e quindi è lasciato non connesso nel SIMCopier. Il pin CLK (C3), come detto precedentemente, è collegato all'uscita dell'oscillatore e fornisce alla carta a micropro-

cessore il segnale di clock necessario per il corretto funzionamento. A parte i contatti di alimentazione (C1, collegato direttamente a 5V) e di massa (C5, collegato direttamente a massa), rimangono i due pin RST (C2) e I/O (C7). Entrambi vengono utilizzati per lo scambio dei dati con il dispositivo di interfacciamento.
Il primo rappresenta un ingresso per la smartcard e viene utilizzato per generare la condizione di Reset che porta la SIM al suo funzionamento, così come avviene con il Reset di un microcontrollore (non dimentichiamo che, tutto sommato, all'interno di una SIM c'è proprio un microprocessore). Nel SIMCopier il pin RST di entrambe le smartcard è collegato ad un pin del PIC (RB6 e RB3) che, via software, può generare la condizione di Reset.
Il pin I/O è il canale di comunicazione vero e proprio tra smartcard e lettore: attraverso questo contatto, i due dispositivi si scambiano le informazioni. Essendo un unico contatto, esso è bidirezionale ed è anch'esso collegato ad un pin del micro per entrambi i connettori (RB1 e RB7).
I condensatori da 100nF C3, C8, C9, C11 sono utilizzati come vie di fuga per eventuali disturbi o variazioni veloci dell'alimentazione. Per questo sono stati posizionati, sul PCB, il più vicino possibile al componente di riferimento.

FIRMWARE

Ovviamente, quando c'è di mezzo un microcontrollore, non basta disegnare lo schema elettrico, ma è anche necessario scrivere un firmware. Nel caso del SIMCopier, il programma in esecuzione nel PIC dovrà rilevare la presenza delle SIM all'interno dei connettori e aspettare la pressione del tasto.
Successivamente, dovrà trasmettere gli opportuni comandi alle due SIM per effettuare la copia della rubrica, cioè di ogni record presente all'interno del file EF_{ADN}.

Rilevare la presenza di una SIM

Il compito che si prevedeva più semplice è diventato un pò più complicato del previsto. Infatti, inizialmente avevo pensato di rilevare la presenza di una SIM inserita in uno slot, monitorando lo stato dell'interruttore integrato nel connettore.
Dopo qualche tentativo, però, mi sono reso conto che questa idea non era molto funzionale: cosa succede se viene inserita una smartcard diversa da una SIM o, addirittura, l'adattatore per le SIM privo della scheda? Poiché l'interruttore è meccanico, questo rileverebbe comunque la presenza di una SIM! Ho risolto questo problema complicando leggermente questo controllo di rilevazione presenza, così come mostrato nel diagramma di flusso in figura 3.

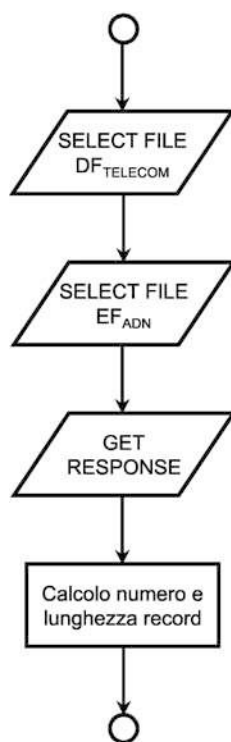



Figura 4 *Diagramma di flusso relativo alla selezione dei file*

Oltre a controllare che l'interruttore sia aperto, cioè che una smartcard (o qualcosa di simile) sia stata inserita nel connettore, il firm-ware prova a generare una condizione di Reset mediante il pin RST. Com'è noto (si veda la parte III del tutorial), una smartcard risponde alla condizione di Reset con l'ATR (Answer To Reset, risposta al Reset).
Controllando la validità di questa risposta, che rispetta certe regole, il microcontrollore può decidere con assoluta certezza se la "cosa" inserita nello zoccolo è effettivamente una SIM correttamente comunicante oppure no. Nel primo caso, viene acceso il led relativo.
Naturalmente questo controllo viene fatto ciclicamente sia per la SIM sorgente che per la SIM destinazione.

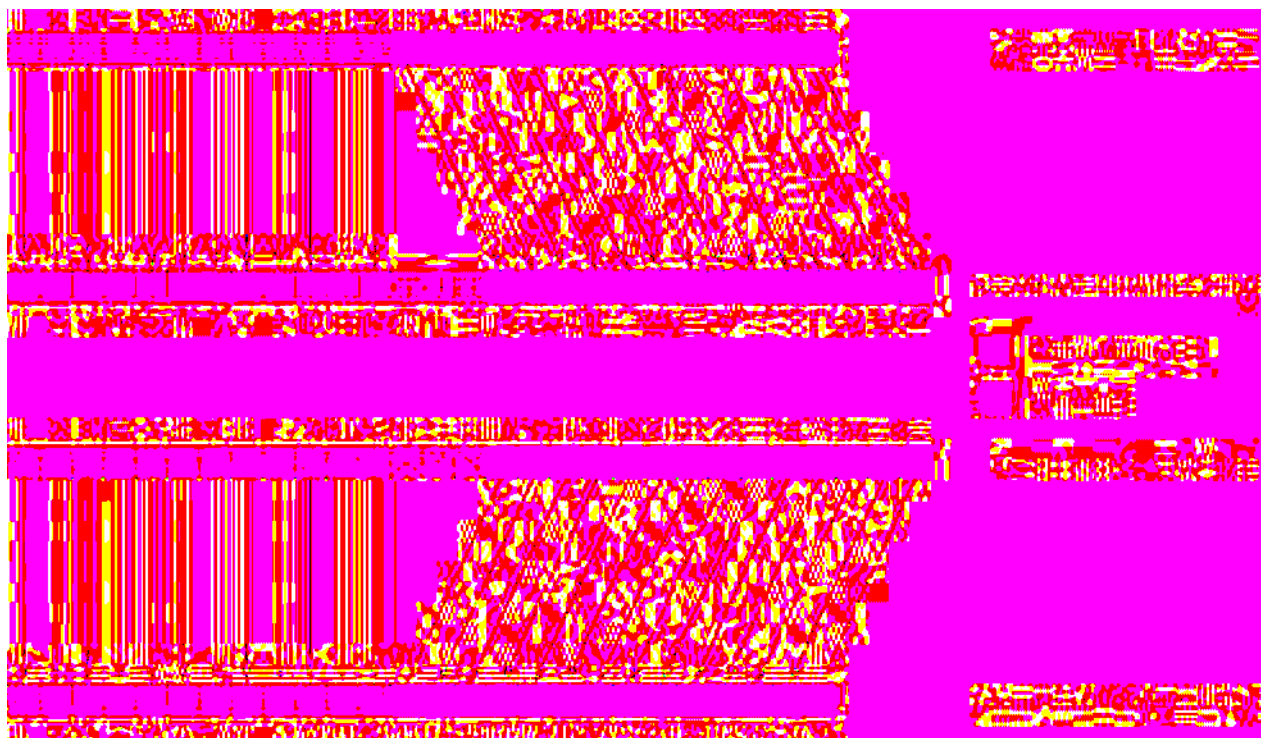



Figura 5 *Esempio di copia di un record tra due SIM di dimensione differente*

Copiare la rubrica

Nella routine principale del firmware, oltre a controllare la presenza delle SIM, viene anche controllata la pressione del tasto che fa partire la procedura di copia della rubrica.
Se il tasto viene premuto e le due SIM sono entrambe presenti nei rispettivi connettori, la copia ha effettivamente inizio.
Come abbiamo già detto all'inizio dell'articolo, la rubrica si trova memorizzata all'interno del file EF_{ADN}, contenuto nel $DF_{TELECOM}$ contenuto, a sua volta, nell'MF.
Prima di poter leggere o scrivere un qualsiasi EF, è necessario selezionarlo mediante l'apposito comando SELECT FILE. Tenendo presente le regole di selezione dei file, per poter selezionare l'EF_{ADN}, è necessario selezionare prima il $DF_{TELECOM}$ (non è necessario selezionare l'MF poiché automaticamente selezionato dopo un Reset). Dopo aver selezionato il file EF_{ADN}, ne viene letto l'header mediante il comando GET RESPONSE.
Analizzando la risposta a quest'ultimo comando, è possibile conoscere numerose

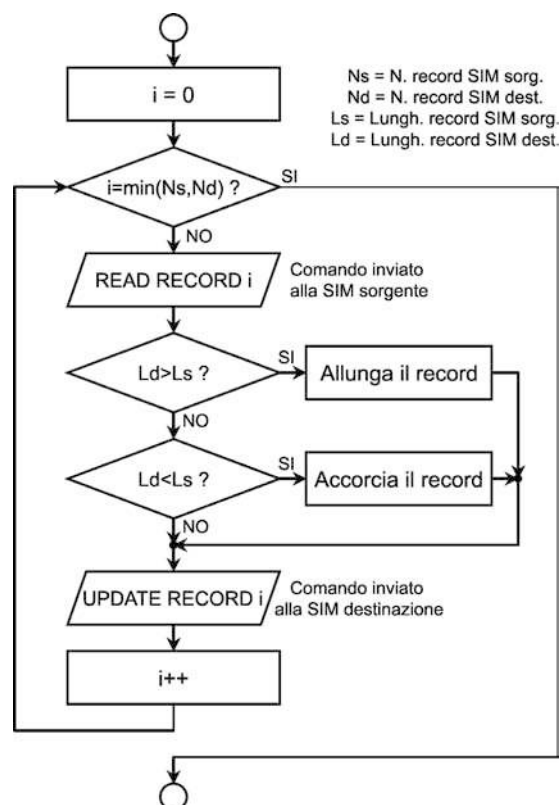



Figura 6 *Diagramma di flusso relativo alla copia dei record della rubrica*

informazioni sul file appena selezionato.
Nel nostro caso, ci interessa sapere il numero dei record presenti nel file e la loro dimensione: infatti, questi due valori possono cambiare a seconda della capacità di memorizzazione della SIM. Per essere precisi, nell'header di un file basato su record vengono indicati il numero dei record e la dimensione totale del file; per conoscere la dimensione di un singolo record, è necessario dividere la dimensione totale del file per il numero dei record.
Gli stessi comandi inviati alla SIM sorgente vengono inviati anche alla SIM destinazione. Questa parte del firmware è rappresentata nel diagramma di flusso in figura 4.

L'ultima parte del programma riguarda la copia di tutti i record dalla SIM sorgente alla SIM destinazione, mediante i comandi READ RECORD e UPDATE RECORD. Si tratta semplicemente di effettuare un ciclo su tutti i record della SIM sorgente, a cui viene inviato il comando READ RECORD ed il comando UPDATE RECORD alla SIM destinazione. Ma anche questa apparente semplice operazione nasconde alcune difficoltà, dovute al fatto che la capacità di memorizzazione delle due SIM può essere diversa.
Prima di tutto, potrebbe essere diverso il numero dei record. È facile trovare in giro SIM che riescono a memorizzare 100 o 250 nomi in rubrica. Cosa succede se due schede diverse vengono utilizzate nel SIMCopier? L'unica soluzione che ho trovato è quella di copiare al massimo il numero di record ospitabili nella SIM destinazione. In altre parole, se la SIM sorgente ha 100 record in rubrica e la SIM destinazione 250, verranno copiati tutti i 100 record nelle prime posizioni della rubrica della SIM destinazione.
Al contrario, se la SIM sorgente memorizza 250 record e la SIM destinazione solo 100, verranno copiati solo i primi 100 record della SIM sorgente!
È come se si volesse copiare il contenuto di un DVD su un CD: al massimo, si potrà solo copiare una parte.
Il secondo problema riguarda la dimensione di un singolo record, anche questa variabile a seconda della capacità di memorizzazione della SIM. Cosa succede se il record della SIM sorgente è più grande o più piccolo di quello della SIM destinazione?
Come già detto precedentemente (vedi figura 1), il campo di dimensione variabile è quello relativo al nome associato al numero telefonico: i restanti 14 byte del record sono fissi e conservano, tra l'altro, il numero telefonico. Per poter adattare un record dalla SIM sorgente alla SIM destinazione, è possibile manipolare solo il primo "campo anagrafico". Così, se il record della SIM sorgente è più piccolo, verrà allungato lo spazio relativo al nome aggiungendo dei byte di riempimento (FF in esadecimale, così come da normative). Al contrario, se il record sorgente è più grande del record destinazione, il nome verrà opportunamente troncato.
Questa operazione è illustrata schematicamente in figura 5, nel caso di un record di dimensione pari a 26 e 29 byte (12 e 15 caratteri per nomi).

Quest'ultima parte del firmware è rappresentata sotto forma di diagramma di flusso in figura 6.

Alla fine dell'operazione di copia, verrà acceso solo il led verde di DL4 per mostrare l'avvenuta copia. Se, per un qualsiasi motivo, c'è un problema durante la copia dei record, il firmware si blocca e viene acceso il solo led rosso di DL4 ad indicazione dell'errore.

REALIZZAZIONE PROTOTIPO

Montaggio

Ho sbrogliato lo schema elettrico precedentemente descritto su un circuito delle dimensioni di 10x8 cm (mezza eurocard). In figura 7 è mostrato il solo lato rame con le piste, le piazzole ed i fori in scala 1:1.
Utilizzando questa figura, potete realizzare il circuito stampato del SIMCopier, avvalendovi del solito sistema di fotoincisione o di altro equivalente. Una volta realizzato il circuito

Elenco componenti	
Sigla	**Valore**
C1, C2	Condensatore 22pF
C3, C4, C5, 8, C9, C11	Condensatore 100nF
C6, C7	Condensatore elettrolitico 10µF
C10	Condensatore elettrolitico 1µF
D1	Diodo 1N4148
D2	Diodo 1N4004 o equivalente
DL1, DL2, L3	Diodo led verde 3mm
DL4	Diodo led bicolore rosso-verde 3mm
IC1	Microcontrollore PIC16F628A
IC2	74HC00
IC3	7805
Q1	Quarzo 3.579MHz
R1	Resistenza 1MΩ
R2	Resistenza 4.7kΩ
R3, R4, R5, R6, R7	Resistenza 330Ω
R8	Resistenza 100Ω
S1	Deviatore per circuito stampato a levetta
T1	Tasto momentaneo due deviatori 4 contatti
J1	Connettore per alimentazione
J2, J3	Connettore ISO7816 con interruttore normalmente chiuso

Tabella 2 *Elenco componenti*

stampato, potete passare al montaggio dei vari componenti, aiutandovi con lo schema di montaggio illustrato in figura 8 e con l'elenco componenti in tabella 2.

Il SIMCopier non presenta particolari difficoltà di montaggio. Iniziate come al solito con i componenti a basso profilo: ponticelli, resistenze, diodi, zoccoli e connettori. Per una pura questione estetica, ho preferito localizzare tutti i ponticelli sotto altri componenti in modo che potessero essere nascosti.
È consigliabile saldare subito i ponticelli e successivamente i componenti che li nascondono. Ce ne sono tre sotto lo zoccolo della SIM sorgente (J3), quattro sotto lo zoccolo della SIM destinazione (J2) ed uno sotto il microcontrollore (IC1). Proseguite con i componenti a più alto profilo, quali i condensatori, i led, il quarzo, il 7805 e il deviatore dell'alimentazione.
Infine, inserite l'integrato 74HC00 e il PIC16F628 nei rispettivi zoccoli.
Ovviamente il PIC va precedentemente programmato con l'apposito file eseguibile che trovate sul sito di Fare Elettronica nelle pagine relative a questo articolo. Il file eseguibile si chiama SIMCopier.hex che potete utilizzare con il software che normalmente usate per la programmazione dei PIC.
La parola di configurazione è memorizzata all'interno del file eseguibile, quindi non dovreste fare nulla di più che caricare il file e scaricarlo nella memoria Flash del microcontrollore. Nel caso il vostro programma non legga correttamente la parola di configurazione all'interno del file eseguibile, dovete abilitare i bit PWRT (PoWeR up Timer), BODEN (Brown Out Detect ENable) e MCLRE (Master Clear Enable), impostare come funzionamento dell'oscillatore un clock esterno (EC, External Clock) e scegliere come volete i bit per la protezione dalla lettura del codice. In figura 9 è mostrata la finestra di ICProg, un noto software freeware utilizzato per la programmazione dei PIC, in cui sono mostrati anche i bit di configurazione.

Alla fine del montaggio, vi consiglio di etichettare i due connettori per le SIM come SOURCE (sorgente) e DESTINATION (destinazione), in modo che non possiate confonder-

li. Tenete presente che, se invertite le SIM, otterrete la sovrascrittura, quindi la perdita, della SIM che, invece, volevate duplicare.
Nel mio prototipo ho applicato una etichetta anche al tasto per indicarne il funzionamento. Il risultato è quello che vedete in figura 10.

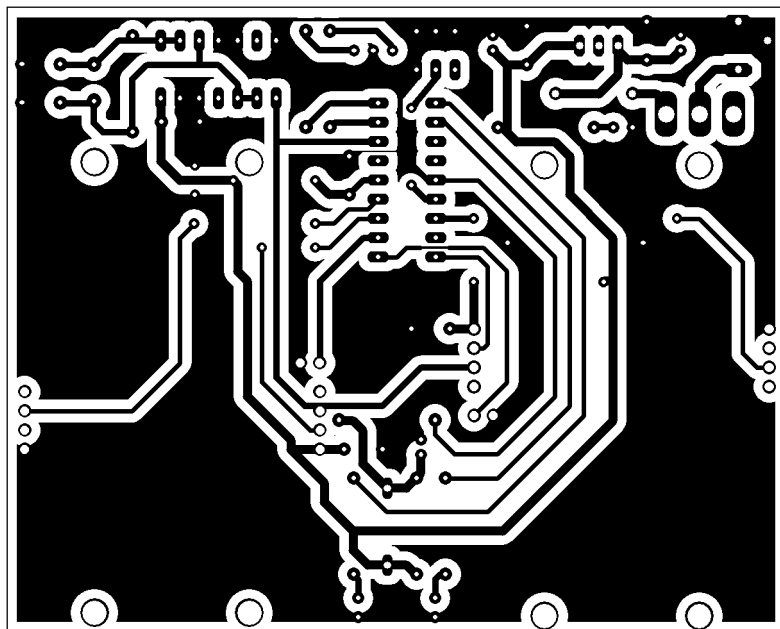



Figura 7 *Circuito stampato in scala 1:1 (lato rame)*

Figura 8 *Piano di montaggio*

Funzionamento

Il funzionamento del circuito è molto semplice ed è stato già descritto tra le righe.
Non appena alimentate il circuito, dovrà accendersi il led verde relativo all'alimentazione, indicando la corretta presenza della tensione 5V. Se inserite una SIM all'interno dei connettori ISO7816, il led relativo si accenderà ad indicare che il microcontrollore comunica correttamente con la scheda.
Se provate ad inserire una carta a banda magnetica o una smartcard diversa da una SIM, il microcontrollore non riuscirà a comunicare con la scheda ed il led rimarrà spento. Ricordate di disabilitare il PIN su entrambe le SIM prima di inserirle nel SIMCopier. In caso contrario, i led di stato si accenderanno ugualmente, ma la procedura di copia terminerà con un errore.
Dopo aver inserito le due SIM e, quindi, dopo l'accensione dei due led di stato, potete iniziare la procedura di copia premendo il tasto. Il led bicolore diventerà giallo durante tutta l'operazione mentre i led di stato delle SIM cominceranno a lampeggiare: ad ogni comandi READ RECORD verrà acceso il led relativo alla SIM sorgente, mentre ad ogni comando UPDATE RECORD verrà acceso il led relativo alla SIM destinazione.
Dopo qualche decina di secondi, la copia termina, lasciando i due led di stato delle SIM spenti e il led bicolore acceso: rosso (copia fallita) o verde (copia effettuata con successo).
A questo punto, potete spegnere il SIMCopier ed inserire le SIM in un telefonino per verificare l'av-

Figura 9 Finestra del programma ICProg

venuta copia. Se è il caso, ricordate di riabilitare il PIN.

STANCHI DI GIOCARE?

Faccio notare che la SIM destinazione può essere tranquillamente una vecchia SIM che avete dimenticato da qualche parte e che è stata disabilitata dal gestore telefonico.
Non è importante, ai fini della copia della rubrica, che la SIM sia abilitata alle telefonate, poiché il contenuto del file della rubrica può comunque essere sempre letto e/o scritto.
E allora perché non utilizzare quella vecchia SIM buttata in un cassetto come backup della vostra rubrica? Questa semplice operazione potrebbe risultare molto utile nel caso in cui smarrite o distruggete la SIM che utilizzate.
È bene precisare che il SIMCopier duplica la rubrica presente all'interno di una SIM e non quella eventualmente presente nella memoria del telefono.

Figura 10 Foto del prototipo

Sviluppi futuri

Il SIMCopier è nato solo per curiosità e non si può certo ritenere un prodotto completo o paragonabile ad apparati commerciali che fanno la stessa cosa. Chi volesse ampliarne le caratteristiche potrebbe pensare di aggiungere un tastierino numerico ed un display. In questo modo, sarebbe possibile inserire direttamente il PIN delle SIM senza doverlo disabilitare prima di utilizzarle.
Inoltre, il display permetterebbe di avere maggiori informazioni sullo stato di avanzamento della copia: per esempio, è possibile visualizzare il numero dei record di entrambe le SIM, la lunghezza di un nome in rubrica, eventuali errori, ecc.
Un'altra possibilità molto interessante è quella di copiare non soltanto il contenuto del file EF_{ADN}, ma anche quello del file EF_{SMS} che conserva gli SMS.

Ringraziamenti

Desidero ringraziare l'amico Giuseppe de Gennaro per il grande aiuto che mi ha dato durante lo sbroglio e la realizzazione del prototipo.

BIBLIOGRAFIA

ETSI TS 100 977: *Specification of the SIM-ME Interface.*
ETSI GSM 02.17: *Subscriber Identity Modules, Functional Characteristics*
ISO7816 parti 1, 2, 3, 4.

RABBIT

Connetti le tue idee con le nostre

RABBITCORE 2xxx / 3xxx

"Schede a microprocessore per abbattere il time-to-market basate su Rabbit2000 e Rabbit3000"

- Fino a 512KB di flash e 512KB di SRAM
- Interfaccia enthernet 10baseT integrata

Time to market

Micro processore 8 bit

- veloce
- potente
- enthernet/internet ready

RABBIT 2000

- 5,5V
- Clock a 30 MHz
- 40 I/O digitali
- 4 porte seriali

SW Royalties free

DEVELOPMENT KIT

"Completi ed a basso costo"

- Librerie a livello sorgente incluse Stack TCP/IP royalties free

LOW COST da 139$

new EMI free

RABBIT 3000

"Il microprocessore EMI free"

- 1,5 - 3,6V (5V tolerant I/O)
- Clock a 54 MHz
- 56 I/O digitali
- 6 porte seriali a interfaccia IrDA

Per ulteriori informazioni

e-mail: informazioni@comprel.it

Oppure visita il sito: http://www.comprel.it

Puoi anche contattarci ai seguenti numeri: Tel.:0362 4961 - Fax: 0362 496433

Via Saragat, 4 - 20054 Nova Milanese (MI)

L'amplificatore operazionale dalla A alla Z

Si esaurisce qui la parte dedicata all'esposizione dei parametri fondamentali di un amplificatore operazionale. Dal prossimo articolo le prime applicazioni pratiche.

3.1 ANALISI DELLA RISPOSTA IN FREQUENZA PER L'OP-AMP AD ANELLO APERTO

Più volte, nei precedenti articoli, si è fatto riferimento alla curva di risposta in frequenza (diagramma di Bode) riportata nella figura 3.1. Questa curva rappresenta la classica risposta di un filtro passa-basso che si definisce del primo ordine dal momento che l'attenuazione, a valle della frequenza di taglio f_t, ha pendenza di 20 dB per decade. Con riferimento all'amplificatore operazionale ciò significa che il guadagno, oltre la frequenza f_t, decresce di 20 dB per ogni decuplicazione della frequenza.
Per ricavare l'espressione del guadagno al variare della frequenza conviene fare riferimento al modello della figura 3.2 nella quale l'amplificatore operazionale si assimila a un circuito con resistenza di ingresso R_{in} molto elevata, e a un generatore di tensione $A_{OL}\ V_i = A_{OL}\ (V_a - V_b)$ di resistenza R_o. L'amplificatore operazionale amplifica infatti la differenza delle tensioni applicate ai suoi due ingressi e pertanto per il guadagno A_{OL}, da intendersi sempre come rapporto fra la tensione V_o di uscita e la tensione di ingresso $V_i = V_a - V_b$, si ha:

$$A_{OL} = V_o / V_i = V_o / (V_a - V_b)$$

La capacità C posta in uscita indica l'insieme delle capacità interne all'op-amp, capacità fondamentalmente dovute alla struttura tecnologica dello stesso operazionale e determinanti al fine dell'andamento della curva di risposta.
Considerando, pertanto, il circuito R_o-C di uscita si può scrivere:

$$V_0 = A_{OL}\ V_i \cdot \frac{\dfrac{1}{j\,2\,\pi\,f\,C}}{R_0 + \dfrac{1}{j\,2\,\pi\,f\,C}} = \frac{A_{OL}\ V_i}{1 + j\,2\,\pi\,f\,C\,R_0}$$

Ma poiché, $(2\ \pi\ R_o\ C) = 1/f_t$, e $V_o\ /\ V_i = A_{OL}$, dall'espressione precedente si ricava:

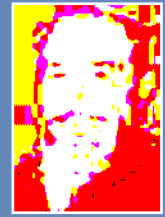
$$A_{OL}\ (f) = \frac{A_{OLo}}{1 + j\,\dfrac{f}{f_t}} \qquad [3.1]$$

dove $A_{OL}\ (f)$ è il guadagno ad anello aperto, mentre A_{OLo} è il guadagno dell'operazionale in continua, ossia per $f = 0$.
Si constata quindi come il guadagno A_{OL} sia funzione complessa della frequenza e come sia pertanto dotato di modulo e argomento espressi rispettivamente dalle equazioni:

$$A_{OL}\ (f) = \frac{A_{OLo}}{\sqrt{1 + \left(\dfrac{f}{f_t}\right)^2}} \qquad [3.2]$$

$$\varphi = -\ arctg\ \frac{f}{f_t} \qquad [3.3]$$

La [3.2], se espressa in dB, diviene:

di Nico Grilloni
n.grilloni@farelettronica.com

La risposta in frequenza dell'op-amp open loop

$$A_{OL}\,(dB) = 20 \log A_{OLo} - 20 \log \sqrt{1 + \left(\frac{f}{f_t}\right)^2} \quad [3.4]$$

Se, per esempio, si considera la curva di risposta della figura 3.1 relativa all'op-amp LM833 che ha un guadagno ad anello aperto A_{OLo} = 113 dB ($10^{5,65}$) e una frequenza di taglio f_t = 28,6 Hz, alla frequenza $f = f_t$ dalla [3.4] si ricava:

$$A_{OL}\,(dB) = 20 \log 10^{5,65} - 20 \log\sqrt{1+1} = 110\ dB$$

Questo risultato era scontato dal momento che per una frequenza f coincidente con la frequenza di taglio f_t intrinseca dell'op-amp, il guadagno deve ovviamente essere di 3 dB inferiore al guadagno massimo. Per cui, essendo quest'ultimo pari a 113 dB, a –3 dB si dovevano necessariamente ottenere 110 dB.
Analogamente, per una frequenza $f = 4\ f_t = 4 \times 28{,}6 = 114{,}4$ Hz, dalla [3.4] si ricava:

$$A_{OL}\,(dB) = 20 \log 10^{5,65} - 20 \log\sqrt{17} = 100{,}7\ dB$$

e per $f = 40\ f_t = 1144$ Hz

$$A_{OL}\,(dB) = 20 \log 10^{5,65} - 20 \log\sqrt{1601} = 80{,}9\ dB$$

Gli ultimi due calcoli evidenziano che la pendenza della curva è di 20 dB per decade.
Infatti, passando da $f = 4\ f_t$ a $f = 40\ f_t$, ossia nell'intervallo di una decade, il guadagno è diminuito di 20 dB, essendo passato da 100,7 dB a 80,9 dB (esattamente è diminuito di 19,8 dB).

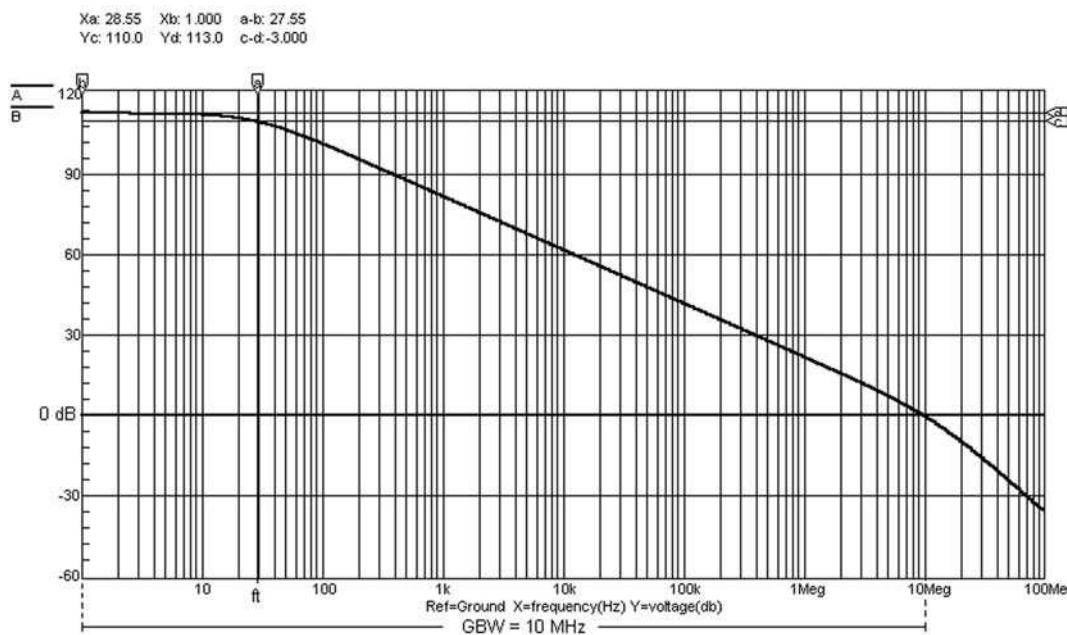



Figura 3.1 *Curva di risposta in frequenza dell'amplificatore operazionale LM833 nella configurazione ad anello aperto. La frequenza di taglio ft, indicata dal marker verticale a, è di circa 30 Hz (Xa = 28,55 Hz)*

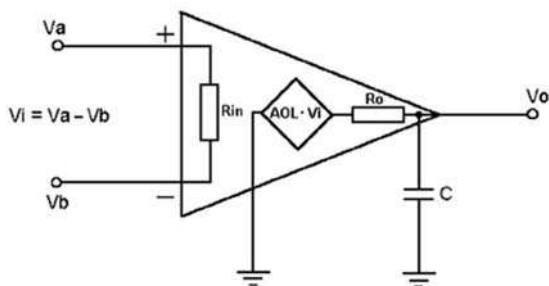



Figura 3.2 *Modello a cui può ricondursi l'amplificatore operazionale classico, anche definito op-amp a reazione di tensione*

L'equazione [3.4] può quindi essere molto utile, oltre che per ottenere per punti la curva di risposta, ancor più per sapere qual è il guadagno in corrispondenza di una determinata frequenza o, viceversa, per sapere qual è la frequenza di taglio in corrispondenza di un determinato guadagno.

Per quanto riguarda la fase, dall'espressione [3.3], per $f = 4\ f_t = 114{,}4$ Hz, per esempio, si ha:

$$\varphi = -\ \text{arctg}\ (4\ f_t / f_t) = -\ \text{arctg}\ 4 = -\ 75{,}96°$$

e per $f = 40\ f_t = 1144$ Hz si ha

$$\varphi = -\ \text{arctg}\ (40\ f_t / f_t) = -\ \text{arctg}\ 40 = -\ 88{,}56°$$

Questi risultati indicano che il segnale di uscita V_o è in ritardo di 75,96° alla frequenza di 144,4 Hz e di 88,56° alla frequenza di 1144 Hz. La figura 3.3 riporta il diagramma di fase ricavato in simulazione. Il marker verticale b indica la frequenza f = 114,4 Hz. Il marker orizzontale *d* interseca il marker *b* e la curva a una distanza $Yd = -75{,}9°$.
L'ordinata *Yd* indica lo sfasamento di 75,9° in ritardo della V_o di uscita rispetto alla V_i di ingresso. Analogamente, il marker verticale a indica la frequenza f = 1144 Hz. Il marker orizzontale *c* interseca il marker verticale *a* e la curva a una distanza $Yc = -88{,}56°$.
L'ordinata *Yc* indica lo sfasamento di 88,56° in ritardo della V_o di uscita rispetto alla V_i di ingresso. Dallo stesso diagramma si vede che lo sfasamento fra segnale di ingresso e segnale di uscita – uscita in ritardo sull'ingresso – può essere tutt'al più di 90°.

3.2 ALTRI PARAMETRI

I dati tecnici relativi agli amplificatori operazionali forniscono, oltre alle grandezze su esposte, anche i valori di altri parametri strettamente inerenti la prevalente destinazione d'uso del componente.

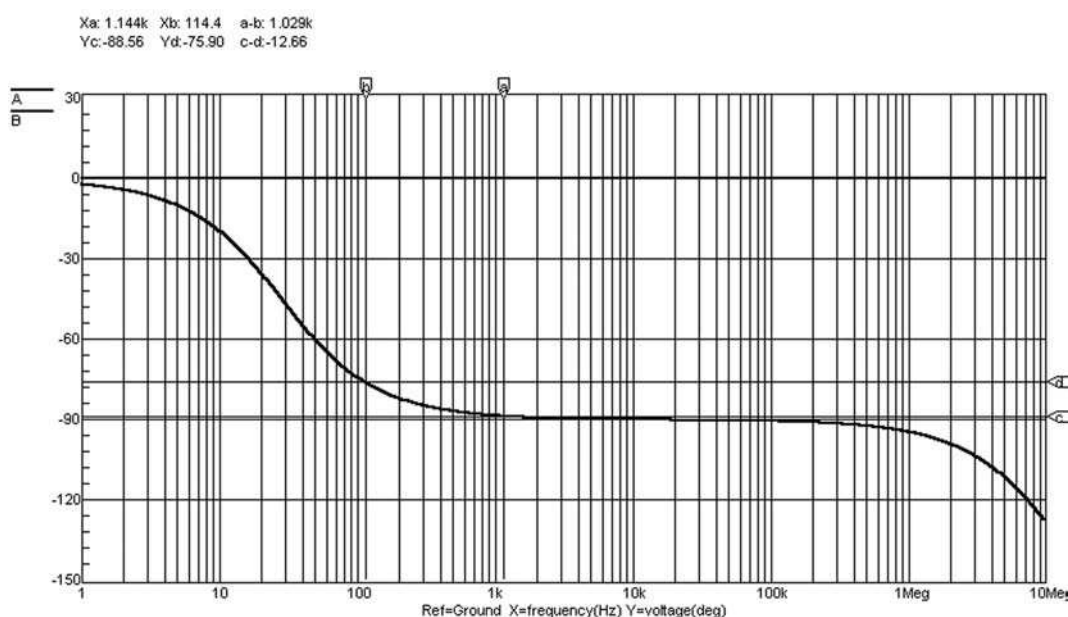



Figura 3.3 *Andamento della fase in funzione della frequenza nell'LM833 nella configurazione ad anello aperto*

Per esempio, nel caso dell'LM833 che, come si è detto, in linea di massima è destinato all'utilizzazione in sistemi audio Hi-Fi, vengono riportati anche i valori del tasso di distorsione armonica totale espresso percentualmente, e la *Power Bandwidth* (quest'ultima indica la massima frequenza alla quale si può avere in uscita un valore di tensione di 10 V di picco senza distorsione).

3.2.1 Tensione e corrente di rumore

La tensione di rumore (*Input Referred Noise Voltage*), di cui i manuali forniscono il valore efficace (o il valore quadratico medio), è misurata alla frequenza di 1 kHz. Il suo andamento presenta valori più elevati mediamente fra 10 e 100 Hz, mentre da 100 Hz in poi si mantiene costante.
Per l'LM833 la tensione di rumore ha un valore tipico di 4,5 nV√Hz oltre i 150 Hz. La corrente di rumore (*Input Referred Noise Current*), sempre espressa in valore efficace (o in valore quadratico medio), è anch'essa misurata alla frequenza di 1 kHz e, al pari della tensione di rumore, presenta, in funzione della frequenza, un andamento piatto al di sopra dei 100 Hz. Per l'LM833 ha un valore tipico di 0,7 pA√Hz oltre i 100 Hz.
Per bassi valori della resistenza del generatore di segnale gli effetti della corrente di rumore possono ritenersi trascurabili ed è prevalente, come fonte di rumore, la tensione di rumore. Per alti valori della resistenza della sorgente diviene invece prevalente, come fonte di rumore, la corrente di rumore. Questo il motivo per cui alcune Case forniscono il valore della tensione di rumore per bassi valori della resistenza della sorgente e la corrente di rumore per alti valori della medesima resistenza.
Gli amplificatori operazionali con ingresso a BJT presentano una tensione di rumore più bassa per valori della resistenza della sorgente di segnale mediamente inferiore a 10 kΩ, mentre gli amplificatori operazionali con ingresso a Fet presentano una minore corrente di rumore che diviene significativa solo se la stessa resistenza è più elevata di 10 kΩ.
L'andamento del rumore è del tutto casuale poiché del tutto casuali sono le relative sorgenti. Queste, prendendo in considerazione l'amplificatore operazionale inserito in circuito, possono essere sia interne alla configurazione circuitale (cablaggio non idoneo con molteplici ritorni a massa, alimentazione non sufficientemente filtrata e non bypassata da condensatori di opportuno valore e/o non cablati vicinissimi ai pin di alimentazione, distanza eccessiva fra stadio alimentatore e circuito con op-amp), sia esterne (segnali a radiofrequenza in aria, induzione del campo magnetico creato dalla frequenza di rete o da altre sorgenti, azionamento di macchine elettriche in prossimità del circuito, ecc.).
Per le cause esterne si può comunque provvedere tramite efficaci schermature che sottraggano il circuito con op-amp da qualsiasi fenomeno di origine elettromagnetica, mentre per le cause interne, oltre a provvedere ad un adeguato cablaggio dei vari componenti, si può quantomeno minimizzare il rumore anche attraverso un'oculata scelta dei componenti passivi a contorno dell'op-amp, dal momento che anche questi costituiscono una fonte di rumore.

3.2.2 La capacità di ingresso (Input Capacitance)

La capacità di ingresso, misurabile fra uno qualsiasi dei due ingressi e massa mentre l'altro ingresso è a massa, anche se non sempre riportata nelle note tecniche, è un parametro importante dal momento che determina una diminuzione dell'impedenza di ingresso dell'operazionale all'aumentare della frequenza. Infatti il classico comportamento della reattanza capacitiva fa sì che questa capacità, a frequenze molto levate, determini un effetto *shunt* sull'ingresso dell'op-amp. Il suo valore è dell'ordine di qualche unità di pF.

3.2.3 Separazione fra i canali

Questo parametro, fornito, ma non sempre, quando nel singolo chip ci sono due o più amplificatori operazionali, esplicita la misura della reciproca influenza degli op-amp medesimi. In Hi-Fi, per esempio, per un ascolto stereo-

fonico apprezzabile è necessario che ci sia una buona separazione fra il canale destro e il canale sinistro. Ove ciò venga a mancare l'ascolto è monofonico e non stereofonico: la differenza con la monofonia pura risiede soltanto nel fatto che i suoni provengono da due canali distinti e non da uno soltanto. Poiché gli op-amp sono nel medesimo involucro, pur inviando il segnale all'ingresso di un solo operazionale ci sarà sempre sull'uscita degli altri una quota parte del segnale. Tanto più basso sarà il livello di quest'ultimo rispetto al livello del segnale di uscita dell'op-amp pilotato, tanto maggiore sarà la separazione fra i canali che, in applicazioni Hi-Fi, come ordine di grandezza non dovrebbe mai essere inferiore a 60 dB. Con riferimento, per esempio, a due dei quattro op-amp interni all'LM124, per il quale la Casa indica una separazione di –120 dB, se V_o1 e V_o2 sono i livelli di uscita, rispettivamente dell'operazionale pilotato e del non pilotato, si ha:

$$-120\ \text{dB} = 20 \log (V_o2 / V_o1)$$

ossia:

$$V_o2 = V_o1 \cdot 10^{-6}$$

Il segnale in uscita dall'op-amp non pilotato avrà ampiezza pari $1/10^6$ rispetto al segnale sull'uscita dell'op-amp pilotato.

3.2.4 Risposta ai transitori (Transient Response)

Per la capacità di risposta ad una sollecitazione impulsiva, per esempio a un gradino di tensione, è importante conoscere il tempo di salita (*rise time*) dell'operazionale.
Ricordando che, in genere, si definisce tempo di salita tr il tempo necessario affinché il segnale si porti dal 10 al 90% della sua massima ampiezza, si ha la relazione:

$$t_r = 0{,}35/GBW$$

Il tempo di salita viene misurato, come lo *slew rate*, impostando l'op-amp per un guadagno unitario e con un carico di qualche centinaio di ohm in uscita. Le Case costruttrici forniscono il valore del tempo di salita prevalentemente per gli amplificatori operazionali destinati a "trattare" segnali impulsivi con variazioni di ampiezza estremamente rapide (segnali a onda quadra e rettangolari in genere).

3.2.5 Tempo di assestamento (Settling Time)

È il tempo che intercorre fra l'applicazione all'ingresso dell'amplificatore operazionale di un segnale con fronte ripido di salita (per esempio, un segnale a onda quadra) e il tempo in cui lo stesso segnale si presenta all'uscita con un'ampiezza prestabilita rispetto all'ampiezza finale.

3.2.6 Distorsione armonica totale (Distortion)

È misurata in un intervallo di frequenze ben definito con carico inserito di valore specificato e a guadagno unitario.
Per l'LM833, per esempio, viene specificata una distorsione dello 0,002 % all'interno della banda audio (20 Hz ÷ 20 kHz) con carico $R_L = 2$ kΩ e tensione di uscita di 3 V efficaci.
Le note tecniche riportano anche il diagramma della distorsione in funzione della frequenza per più di un valore della V_o di uscita e per una prestabilita tensione di alimentazione.

3.2.7 Corrente di cortocircuito (Output Short-Circuit Current)

Indicata con I_{sc}, è la corrente erogata dall'amplificatore operazionale con il terminale di uscita direttamente connesso a massa.
La I_{sc} non sempre viene fornita.
Comunque la maggior parte degli operazionali è dotata di un circuito a limitazione di corrente che inibisce la possibilità di erogare una corrente di cortocircuito distruttiva.

NOTE PER IL LETTORE

I diagrammi delle figure 3 a e 3 b sono stati ricavati al computer col software di simulazione SPICE (*Simulation Program of Integrated Circuits Emphasis*) – CircuitMaker – della Microcode Engineering Inc.

gli appuntamenti

1 2 3 4 5 6 7/8 9 10 11 12

ALCUNE DELLE DATE INDICATE POTREBBERO SUBIRE VARIAZIONI

05-07 Maggio 2005

MICROELETTRONICA — Vicenza (VC)

Microelettronica, una delle manifestazioni più importanti a livello nazionale, da sempre punto di riferimento per il settore dei componenti e della produzione elettronica, nel 2005 si focalizzerà sull'automazione La mostra sarà suddivisa in 9 aree tematiche, tra cui spiccano quella dedicata ai Sensori e ai Bus di campo, quella dedicata alle macchine, all'assemblaggio e alla fluidodinamica (precedentemente indicata come SIAS). All'interno di Microelettronica è previsto un programma di oltre 20 tra convegni e seminari.

LUOGO: Fiera di Vicenza (VC)
ORARI: dalle 9:30 alle 18:00
ORGANIZZATORE: Fiera di Vicenza (www.microelettronicafiera.it)
INGRESSO: gratuito (previa registrazione)

07-08 Maggio 2005

GRANDE FIERA DELL'ELETTRONICA — Forlì (FC)

Quello di Forlì è uno degli appuntamenti "classici" per gli appassionati di mostre mercato.
A Forlì troverete computer, programmi, periferiche, accessori, telefonia fissa e mobile, sistemi di sicurezza, strumenti di misurazione, smart card, decoder, antenne e parabole, schede, apparecchiature per radioamatori, CB, radio d'epoca, pezzi rari, valvole e ricambi.

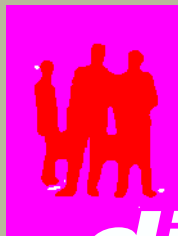
Parallelamente, si svolgono il Flight Simulator Show e il Contest "Fabio Fanelli", due appuntamenti di risonanza nazionale dedicati al volo simulato. Aderiscono all'iniziativa Aziende di altissimo livello internazionale.

LUOGO: Fiera di Forlì - Forlì (FC)
ORARI: dalle 9:00 alle 18:00
ORGANIZZATORE: Blu Nautilus (www.blunautilus.it - Tel. 0541.53294)
INGRESSO: € 7,50 - ridotto € 6,50
vale per entrambe le manifestazioni

14-15 Maggio 2005

12° MARC DI PRIMAVERA — Genova (GE)

Mostramercato attrezzature radioamatoriali e componentistica, ricezione via satellite, informatica, radio d'epoca, editoria specializzata. Come di consueto MARC offre una panoramica completa di attrezzature e componentistica radioamatoriale, ma anche antenne e parabole satellitari, decoder, stampa specializzata e accessori per i telefoni cellulari. Grandissimo risalto ha avuto il settore informatico con un'ampia offerta di pc, notebook, stampanti, scanner, software e videogames. Mentre i cultori del fai-da-te hanno potuto trovare pezzi di ricambio, apparecchiature da mettere a posto e accessori per elaborazioni speciali a costi estremamente competitivi. Ampia l'offerta di radio d'epoca e di apparecchi televisivi anni '50-'70.

LUOGO: Fiera Internazionale di Genova – Genova (GE)
ORARI: dalle 9:00 alle 18:30
ORGANIZZATORE: Studio Fulcro (www.studio-fulcro.it - Tel. 010.561111)
INGRESSO: n.p.

Per sapere a quali appuntamenti sarà presente FareELETTRONICA visita la pagina: www.farelettronica.com/fiere

24-26 Maggio 2005

LINUXWORLD CONFERENCE & EXPO — San Donato Milanese (MI)

LinuxWorld Conference & Expo è il principale forum informativo e formativo dedicato esclusivamente alle soluzioni Linux e Open Source per il mondo business.
L'area espositiva permette ai visitatori di entrare in stretto contatto con le singole realtà industriali operanti nel settore e di toccare con mano le ultime offerte del mercato. Aree speciali come l'Info Arena rappresentano un'opportunità per le società espositrici di presentare le proprie offerte commerciali e le ultime soluzioni per il mercato, mentre il .org Pavilion ospita i principali progetti no-profit del panorama italiano. I convegni e i forum tematici, presieduti dai più competenti esperti italiani e internazionali, consentono di scoprire e condividere esperienze e competenze.

LUOGO: Crowne Plaza Milan Linate, San Donato Milanese (MI)
ORARI: dalle 9:00 alle 18:30
ORGANIZZATORE: Wireless S.r.l. (Tel 02.48100306 www.linuxworldexpo.it)
INGRESSO: gratuito (previa registrazione)

17-21 Maggio 2005

INTEL 2005 — Milano (MI)

19ª Edizione internazionale di elettrotecnica, elettronica, illuminazione, automazione industriale, sicurezza e componentistica. INTEL 2005 è una delle principali rassegne internazionali di elettrotecnica, elettronica, illuminazione, automazione industriale, sicurezza, componentistica, nell'ambito della quale espongono le aziende leader. È un evento suddiviso in cinque aree tematiche sinergiche tra loro molto specializzate e con una forte identità, ciascuna delle quali beneficia di una promozione mirata: World Light Show, Building Show, Power Show, Factory Show, Components Show.

LUOGO: Fiera Milano - Milano (MI)
ORARI: dalle 9:00 alle 18:00
ORGANIZZATORE: Fiera Milano Tech (www.intelshow.com)
INGRESSO: Gratuito (registrazione necessaria)

04-05 Giugno 2005

31° RADIANT AND SILICON — Novegro (MI)

Ospite d'eccezione al 31º RADIANT AND SILICON il razzo "STRATOSFERA". Il razzo esposto non sarà quello definitivo ancora in fase di progettazione, ma il prototipo "Stratosfera Zero" e sarà ubicato presso lo stand dell' A.R.I., sezione di Milano.
Per coloro che non avessero mai visitato RADIANT ricordiamo che gli espositori sono circa 180, distribuiti in tre padiglioni, ed offrono un vasto assortimento merceologico comprendente apparati e componenti per telecomunicazioni, Internet e ricetrasmissioni di terra e satellitari, elettronica, informatica con relativo materiale di consumo, antennistica, telefonia fissa e cellulare, componentistica, kit per autocostruzioni, editoria specializzata ed una Borsa-Scambio di surplus elettronico, informatico ed un'ampia scelta di vecchi apparati radio, di valvole e ricambi.

LUOGO: Parco Esposizioni Novegro – Novegro (MI)
ORARI: dalle 9:00 alle 18:00
ORGANIZZATORE: Comis Lombardia (Tel. 02-7562711 www.parcoesposizioninovegro.it)
INGRESSO: € 8 vale per entrambe le manifestazioni

Prima parte
La programmazione base

Seconda parte
Le variabili, le costanti e gli ingressi

Terza parte
Utilizzare al meglio le uscite del PIC

Mikrobasic per PICmicro

Questo mese approfondiremo l'utilizzo delle variabili e delle costanti in Basic. Vedremo come gestire gli array e come interagire con il microcontrollore attraverso i suoi ingressi. Un altro passo fondamentale nella gestione dei PIC con un linguaggio ad alto livello come il Basic.

LE VARIABILI E LE COSTANTI

Senza le informazioni e i dati da elaborare i programmi non esisterebbero, o meglio non potrebbero produrre i risultati sbalordivi che invece normalmente forniscono. Questi dati risiedono normalmente in alcune celle di memoria. Possono venire gestiti attraverso le variabili e le costanti.

Le variabili

La variabile è semplicemente uno spazio in memoria che può contenere un dato. Come dice il nome, tale dato può essere soggetto a modificazioni e cancellazioni durante l'esecuzione del programma. Possiamo immaginare le variabili come una sorta di scatole, all'interno delle quali si possono inserire dei valori numerici. Siccome solitamente vi sono diverse scatole, è necessario differenziarle tra di loro, semplicemente assegnando un nome che possa distinguerle.
In figura 1 abbiamo due scatole, una piccola

Figura 1 *Concetto di variabile*

ed una grande. La scatola piccola è stata arbitrariamente chiamata ***k***. In realtà possiamo dare alle variabili qualsiasi nome lecito, ma è sempre meglio sceglierne uno appropriato, in base al contenuto.
Dal momento che la variabile k deve "contenere" un numero piccolo (nel nostro caso il numero 3), la relativa scatola deve essere *piccola*. A destra vediamo invece un grande scatolone, cioè la variabile ***persone***, che contiene un grosso numero (il 577). Si deduce quindi che lo spazio preposto a contenere i valori deve essere accuratamente "dimensionato" in base al massimo valore previsto.
Non conviene usare quindi un grosso scatolone se dobbiamo inserirvi solamente un piccolissimo numero.
Non può invece verificarsi il caso contrario, ossia un grosso numero non può entrare in una scatola piccola. Adesso vedremo come si programmano le variabili e come si sceglie la relativa dimensione.
Il Mikrobasic ci mette a disposizione un comando per dimensionare queste "scatole" in base al loro massimo contenuto previsto. Il

Le variabili, le costanti e gli ingressi

di Giovanni Di Maria
g.dimaria@farelettronica.com

Tipi di dato

Tipo	Spazio occupato in memoria	Intervallo di valori
byte	8 bit	0 ÷ 255
char	8 bit	0 ÷ 255
word	16 bit	0 ÷ 65535
short	8 bit	-128 ÷ 127
integer	16 bit	-32768 ÷ 32767
longint	32 bit	-2147483648 ÷ 2147483647

contenuto, come vedremo meglio in seguito, è sempre numerico, anzi in ogni locazione di memoria è sempre contenuto un valore massimo di 255 (ossia un byte). Ecco i tipi di dato che gestisce il compilatore:
Come si vede dalla tabella, a seconda del valore massimo che una variabile assumerà durante l'esecuzione di un programma, occorre usare un tipo dato appropriato. Così ad esempio, se prevediamo che l'indice di un contatore non supererà MAI il valore di 100, utilizzeremo il tipo di dato **byte**. Potremmo usare anche il tipo word o altro più capiente, ma consumeremo inutilmente memoria RAM che, credetelo, è veramente molto limitata. Approntiamo lo schema di cui alla figura 13 presentato nella puntata precedente (FE 237 - Marzo 2005): 8 LED collegati in PORTB.
Esaminiamo il seguente programma che, a dire il vero, potrebbe essere un po' ostico da capire:

```
program scanner

dim k as byte
dim inizio as byte
dim fine as byte

main:
  trisb=0 'PORTB OUTPUT
  portb=0 'Spegne portb
  inizio=30 'Stabilisce punto iniziale
  fine=200 'Stabilisce punti finale
  ciclo:
    for k=inizio to fine
      portb=k 'Visualizza il byte
      Delay_ms(100)  'Aspetta
    next k
end.
```

Si inizia ad informare il compilatore che verranno utilizzate 3 variabili di tipo byte (poiché il loro contenuto, durante l'esecuzione, non supererà il valore 255) tramite l'istruzione **dim**. Dopo aver definito la portb completamente in output (trisb) ed azzerato la stessa, stabiliamo che il conteggio deve iniziare da 30 (variabile *inizio*) e terminare a 200 (variabile *fine*). Il ciclo for farà iterare la variabile k affinché, di volta in volta, assuma il valore sequenziale vincolato da tali limiti. Durante questo lavoro, verrà assegnato alla portb il valore reale della variabile k, comportandosi quindi da monitor di visualizzazione del contenuto della variabile k. Con il termine di assegnazione si intende quella operazione mediante la quale un valore numerico viene "memorizzato" nella variabile.
Un esempio di assegnazione è questo:

```
k = 200
limite = 33000
tempo = 60
gradi = 200
secondi = 30
ore = minuti * 60
```

Come si vede, il valore posto alla destra del simbolo uguale (=) viene assegnato alla variabile di sinistra. Si realizza in altre parole l'immissione di un valore numerico all'interno di una casella di memoria. Le variabili vengono rilasciate, ossia perse, non appena si toglie l'alimentazione al PIC.
La dichiarazione delle variabile va sempre eseguita prima della label ***main***.

Le costanti

A differenza delle variabili, le costanti sono entità il cui contenuto non cambia durante l'esecuzione di un programma.
È quindi utile definirle tali in quanto il compilatore ottimizza il loro utilizzo in termini di spazio occupato e velocità. Quando un'informazione resta immutata nel corso della sua vita (informatica), questa può tranquillamente essere considerata come costante. Nella codifica solitamente è buona norma scriverle in MAIUSCOLO.
Ecco nell'esempio come vanno dichiarate le costanti:

```
program costanti

const NESSUNO     = 0    'Tutti Led spenti
const TUTTI        = 255  'Tutti Led accesi
const PRIMOULTIMO = 129  'Primo e Ultimo ON
const CENTRALI    = 24   'Led centrali ON

main:

  trisb=0 'PORTB in OUTPUT
  portb=0 'Spegne portb

  portb=TUTTI 'Accende TUTTI i LED
  Delay_ms(1000)
  portb=NESSUNO 'Spegne tutti i LED
  Delay_ms(1000)
  portb=PRIMOULTIMO 'Accende primo e ultimo
LED
  Delay_ms(1000)
  portb=CENTRALI 'Accende i due LED centra-
li
  Delay_ms(1000)
  portb=0 'Spegne TUTTO

end.
```

Esaminiamo il software. Dopo la dichiarazione della quattro costanti, e dopo l'inizializzazione della PORTB, tutti i LED si illumineranno per un secondo. Perché?
Perché la costante TUTTI (che vale 255) è assegnata direttamente alla PORTB, e se si converte il numero decimale 255 in notazione binaria, il risultato sarà 11111111, cioè otto cifre abilitate, (otto LED accesi). Effettuiamo adesso la conversione di queste quattro cifre decimali in cifre binarie e vediamo il risultato:

$(255)_{10} = (11111111)_2$
$(0)_{10} = (00000000)_2$
$(129)_{10} = (10000001)_2$
$(24)_{10} = (00011000)_2$

Come si può facilmente notare, le otto cifre rappresentano i nostri otto diodi Led. Se sono tutti accesi è ovvio che stanno visualizzando il byte 255, vale a dire 11111111.
Tutto questo è il *digitale*. L'1 rappresenta il segnale presente, la tensione, la presenza di informazione.
Lo 0 rappresenta il silenzio, assenza di tensione o meglio tensione nulla quindi il Led spento.
Se volete accendere un numero di Led predefiniti dovete convertire sempre la relativa rappresentazione binaria in decimale. Oppure potete abilitare individualmente ogni singola porta, come abbiamo visto nei primi esempi.

Gli array o vettori

Le variabili e le costanti sono oggetti indipendenti l'una dall'altra. Può essere però a volte utile (se non indispensabile) trattarle come membri di una stessa famiglia, e quindi con lo stesso nome, differenziandole solo da un numero progressivo che le accompagni.
Nel precedente programma SuperCar3 abbiamo memorizzato i valori dei Led da illuminare in un vettore di costanti. Esaminiamo attentamente la riga di dimensionamento:

```
const sequenza as
byte[14]=(1,2,4,8,16,32,64,128,64,32,16,8,4
,2)
```

Questo comando crea quattordici costanti di nome ***sequenza***.
Non si tratta di una costante semplice, ma di una famiglia di costanti, tutte chiamate "*sequenza*" ma ognuna "codificata" con un numero progressivo da 0 a 13.
Pertanto l'assegnazione riserverà in memoria ben 14 celle, ognuna delle quali ospiteranno rispettivamente i valori di 1, 2, 4, 8 e così via.
Perché si usano gli array? Non sarebbe più semplice usare le normali variabili o costanti? Ovviamente in questo caso no, poiché avere a disposizione una lista organizzata permette al programma di esaminare i dati più facilmente e sequenzialmente; basta infatti inserire la parte di codice da elaborare all'interno di un efficiente ciclo *FOR-NEXT*.
In questo modo la lunghezza del codice programma è sempre la stessa, indipendentemente dal numero di elementi che compongono il vettore stesso.
Con l'assegnazione di costanti a vettore, il programma si comporta come se ogni elemento singolo fosse così inizializzato:

```
sequenza[0] = 1
sequenza[1] = 2
sequenza[2] = 4
sequenza[3] = 8
sequenza[4] = 16
sequenza[5] = 32
sequenza[6] = 64
sequenza[7] = 128
```

```
sequenza[8] = 64
sequenza[9] = 32
sequenza[10] = 16
sequenza[11] = 8
sequenza[12] = 4
sequenza[13] = 2
```

Con una sola riga di comando abbiamo effettuato tutte le 14 assegnazioni. Non possiamo però modificare il contenuto in quanto si tratta pur sempre di una costante.
Per definire un vettore di tipo "variabile" si usa invece questa codifica:

```
dim vett as byte[4]
```

dove *vett* rappresenta un vettore di quattro elementi di tipo byte, numerati da 0 a 3. In altri termini è come se avessimo quattro variabili: invece di avere nomi diversi hanno un numero di riconoscimento, ossia un *indice*. Come si vede in figura 2, il vettore è un insieme di cassettini, indipendenti l'uno dall'altro, che hanno la possibilità di contenere un valore.
Si usa di solito quando si ha a che fare con molti valori appartenenti alla stessa tipologia di oggetti.

Il precedente programma del Semaforo con le costanti

Vedremo adesso che con l'uso prudente delle costanti la realizzazione del semaforo visto in precedenza risulterà molto più semplice dal punto di vista del software.
Approntiamo nuovamente lo schema della figura 11 presentato nella puntata precedente (FE 237 - Marzo 2005). Scriviamo il programma necessario e poi commentiamo i punti salienti:

```
program Semaforo2

const VERDE =  4 '00000100
const GIALLO = 6 '00000110
const ROSSO  = 1 '00000001

main:
  trisb=0 'Portb OUT
  portb=0 'Spegniamo tutto

  inizio:   'label del ciclo
    portb=VERDE 'Accende verde
    Delay_ms(30000) 'Wait 30 sec.
    portb=GIALLO 'Verde+giallo
    Delay_ms(5000) 'Wait 5 sec.
    portb=ROSSO 'Accende rosso
    Delay_ms(30000) 'Wait 30 sec.
  goto inizio 'Ricomincia tutto
end.
```

Abbiamo utilizzato tre costanti, VERDE, GIALLO, ROSSO. Rispettivamente contengono la notazione binaria della visualizzazione dei led, cioè:

- VERDE vale 4 ed in binario 100 (Verde ON, Giallo OFF, Rosso OFF).
- GIALLO vale 6 ed in binario 110 (Verde ON, Giallo ON, Rosso OFF).
- ROSSO vale 1 ed in binario 001 (Verde OFF, Giallo OFF, Rosso ON).

Le tre costanti vengono quindi rappresentate direttamente dalla PORTB (ricordiamo che questa è un byte a tutti gli effetti), opportunamente ritardate dal comando delay_ms. Il ciclo si ripete all'infinito grazie all'istruzione *goto inizio*.
Bene, proponiamo un ultimo esempio, prima di approfondire altri argomenti.
Analizziamo il seguente problema: abbiamo due diodi LED, rispettivamente di colore rosso e verde, collegati indipendentemente a due

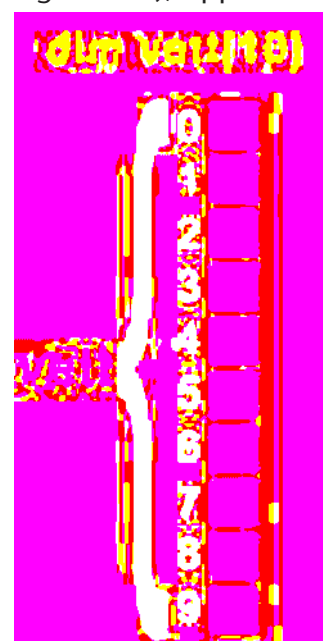



Figura 2 Struttura di un vettore

uscite della PORTB. Alla cadenza di 2 Hz deve lampeggiare il diodo rosso **per dieci volte**, poi alla cadenza di 1 Hz deve lampeggiare il diodo verde **per due volte**. Il ciclo si ripete all'infinito. Realizzate nuovamente il circuito di figura 9 presentato nella puntata precedente (FE 237 - Marzo 2005) rispettando i colori dei diodi LED. Il programma (una delle tante soluzioni possibili) sarà il seguente:

```
program RossoVerde

dim conta as byte

main:
  'Un altro modo di
  'esprimere un valore in
  'codice binario
  'digit per digit

  trisb=%00000000 'PORTB in OUTPUT
  portb=%00000000 'PORTB tutta spenta

  inizio:  'Etichetta di riferimento

  for conta=1 to 10
    portb.0=1 'Rosso ON
    Delay_ms(250)
    portb.0=0 'Rosso OFF
    Delay_ms(250)
  next conta

  for conta=1 to 2
    portb.1=1 'Verde ON
    Delay_ms(500)
    portb.1=0 'Verde OFF
    Delay_ms(500)
  next conta

  goto inizio

end.
```

Nel listato c'è una novità: nell'assegnare il valore 0 ai registri TRIS e PORT si è utilizzato l'operatore %. Collocato all'inizio di un valore, permette la sua codifica in formato binario.

Figura 3 *Temporizzazione di un lampeggio*

Certe volte è usato per abilitare in un solo colpo più uscite. Se ad esempio volessimo accendere (partendo da RB0) la prima, la seconda, la terza e l'ultima porta di PORTB, basterà programmare la porta con la semplice istruzione:

```
PORTB = %10000111
```

RB0 è posto sempre alla destra della cifra binaria (bit meno significativo) mentre RB7 è posto alla sinistra della cifra (bit più significativo).
Analizziamo adesso il problema della temporizzazione. Se il diodo LED rosso deve lampeggiare alla cadenza di 2 Hz, perché abbiamo usato un tempo di attesa di 250 ms. e se il diodo LED verde deve lampeggiare alla cadenza di 1 Hz, perché abbiamo usato un tempo di attesa di 500 ms? Guardiamo la figura 3.
Quando la frequenza, o cadenza, di un lampeggio è, ad esempio, 2 Hz vuol dire che esso si ripete 2 volte al secondo. Pertanto la durata del suo periodo, cioè il tempo che intercorre tra l'inizio dell'evento e l'inizio del successivo evento ammonta a 0,5 secondi. La relazione è data dalla formula $t = 1/f$, dove t è il tempo in secondi ed f è la frequenza. Dal momento che l'intero ciclo dura 0,5 secondi è ovvio che un semiciclo (rappresentato dal led acceso) dura 0,25 secondi (cioè 250 ms.) e l'altro semiciclo (rappresentato dal led spento) dura anche 0,25 secondi.

E ADESSO USIAMO GLI INGRESSI

Sino ad ora abbiamo esaminato solamente il modo in cui il PIC comunica con il mondo

esterno. Adesso vedremo il contrario, cioè come è il mondo esterno a comunicare con il PIC. Si considerano esterni tutti quegli eventi provenienti da fonti non appartenenti alla circuiteria del microcontrollore stesso, come ad esempio l'impulso proveniente da un interruttore o da un pulsante, il comando proveniente da un sensore, le cifre generate da una tastiera esterna, e così via.

Occorre prima di tutto puntualizzare un concetto importantissimo: dal momento che il PIC lavora con tensioni logiche positive (0 Volt e 5 Volt rispetto massa), occorre che anche i segnali provenienti dall'esterno siano adeguati a questo range. Se da un pulsante preleviamo una tensione di 15 Volt, non dovremmo assolutamente applicarla al controllore, pena la sua distruzione.

Occorre quindi che tutti i livelli superiori a 5 Volt siano "accuratamente" adeguati al limite massimo sopportato dal PIC (+/- 10% di tolleranza massima). Sarà utile a tale scopo usare i partitori resistivi affinché la tensione di ingresso si abbassi al valore consentito.

Per il resto si segue la solita storia: un livello in entrata messo a 0 volt rispetto massa sarà considerato dal PIC come un segnale *basso* o *falso* (0), viceversa un segnale di 5 Volt sarà interpretato come un segnale logico *alto* o *vero*.

Consideriamo un altro fatto importante. Tutti gli ingressi devono essere comunicati al PIC in quanto tali. Occorre a tale scopo programmare in anticipo la funzione di ogni pin, ossia se esso fungerà da input oppure da ouput.

Se ad esempio su RB0 colleghiamo un LED e su RB1 colleghiamo un pulsante, in fase di programmazione dobbiamo comunicare al compilatore (attraverso il registro TRIS) che il pin RB0 è un'uscita e quindi un Output (valore **0**) ed il pin RB1 un ingresso e quindi un Input (valore **1**).

Passiamo subito alla pratica.

Occorre creare il seguente prototipo: su RB0 c'è collegato un LED rosso, su RB1 un pulsante normalmente aperto e su RB2 un altro pulsante normalmente aperto. Il circuito deve comportarsi nel modo seguente:

- All'accensione del circuito entrambi i LED sono spenti.
- Se si preme il primo pulsante (su RB1) il LED dovrà illuminarsi e dovrà mantenere tale stato anche al rilascio del pulsante stesso.
- Se si preme il secondo pulsante (su RB2) il LED dovrà spegnersi e dovrà mantenere tale stato anche al rilascio del pulsante stesso.

Non preoccupatevi, è più difficile da leggere che da realizzare. Realizzate lo schema della figura 4, poi passiamo ai commenti elettronici ed informatici. Guardando lo schema si nota che il Led è collegato, tramite la solita resistenza di limitazione di corrente, al pin RB0; quest'ultimo sarà quindi considerato come Output. Sui pin RB1 e RB2 sono collegati invece due pulsanti normalmente aperti.

Ecco il funzionamento del collegamento: se un pulsante è aperto (cioè non premuto) esso non si collega ai 5 Volt dell'alimentazione, pertanto RB1 (o RB2) sarà connesso a massa tramite la resistenza di 1K (resistenza di *pull-down*). In altre parole tale resistenza forza lo stato logico basso (0V) all'ingresso RB1 (o RB2). Viceversa, quando il pulsante è premuto, il pin RB1 (o RB2) si connetterà direttamente ai 5 Volt dell'alimentazione. Il pin passerà pertanto a livello 1 logico e la

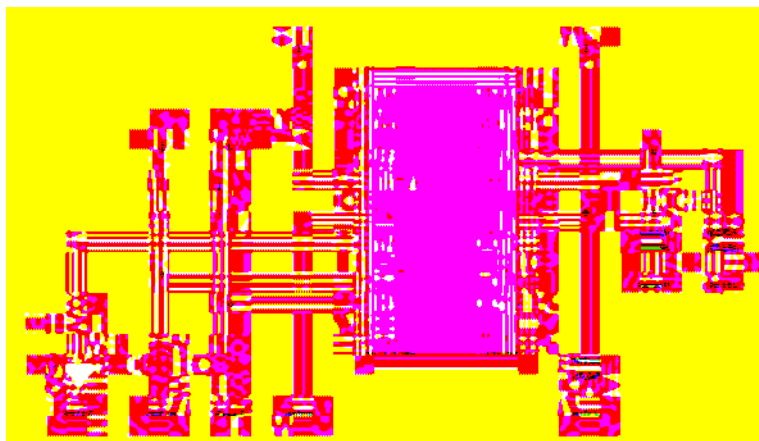



Figura 4 Schema con un LED e due pulsanti (I/O)

resistenza di pull-down non costituirà minimamente un problema, in quanto su essa scorrerà una corrente di soli 5 mA (essendo l'ingresso stesso ad alta impedenza). Volendo "consumare" meno corrente, potremo benissimo "alzare" il valore di queste resistenze fino a 47 Kohm senza problemi. L'importante è che ci siano, altrimenti, data l'alta impedenza dei suoi terminali, il PIC potrebbe leggere e recepire il segnale logico in maniera inaspettata e ambigua (ossia presenza di 1 e 0 senza apparenti spiegazioni "logiche"). Ecco come il PIC riconosce gli eventi esterni: con segnali logici 0-1 che corrispondono a due livelli analogici 0 Volt e 5 Volt. Il problema sarà adesso quello di istruire il microcontrollore a ricevere questi segnali, ad elaborarli e a comportarsi di conseguenza.
Studiamo il relativo programma, leggendo attentamente anche i commenti:

```
program Input1

main:
  'definiamo RB7-6-5-4-3
  'come uscite, RB2 e RB1
  'come entrate e RB0 come
  'uscita. Comunque le porte
  'da RB7 a RB3 in questo
  'esempio restano inutilizzate

  trisb=%00000110

  'In maniera equivalente
  'potremo scrivere:
  'trisb.0=0 (output)
  'trisb.1=1 (input)
  'trisb.2=1 (input)

  'Azzeriamo adesso la SOLA
  'porta in OUTPUT (RB0)
  'Non settiamo MAI da
  'software i pin configurati
  'come input. Potrebbe essere
  'pericoloso (per il PIC).

  portb.0=0 'Spegniamo LED
```

```
  ciclo:    'Label di iterazione

    if portb.1=1 then 'Se premo pulsante 1
      portb.0=1       'accende LED
    end if

    if portb.2=1 then 'Se premo pulsante 2
      portb.0=0       'spegne LED
    end if

  goto ciclo

end.
```

Come si vede, il listato è molto breve, abbiamo prolungato un po' il commento, cioè la parte descrittiva del programma che il compilatore non considera durante la traduzione in linguaggio macchina.
Vediamo come si comporta il software: dopo le definizioni di rito che comandano il funzionamento della porta e il suo azzeramento, si passa direttamente ad esaminare lo **stato logico** delle due porte-ingresso.
Questa operazione viene svolta dal comando IF-THEN che in italiano vuol dire SE-ALLORA. Il micro inizialmente si domanda SE lo stato della RB1 (dove è montato il primo pulsante) si trova a livello logico alto (1). Se così è, viene eseguita subito la successiva istruzione (portb.0=1) cioè effettua l'accensione del led su RB0 (o portb.0, che poi è la stessa cosa). Se invece il livello su RB1 è basso (pulsante non premuto) la successiva istruzione (sempre portb.0=1) non viene minimamente considerata e quindi non è eseguita.
Continuando, il micro si pone usa seconda domanda, cioè se lo stato logico presente su portb.2 (RB2) è alto (pulsante premuto). Se così è, viene eseguita la successiva istruzione (portb.0=0) e il LED si spegne sino ad altra comunicazione. Con il comando *goto ciclo*, l'intero processo si ripete all'infinito.
Abbiamo realizzato una sorta di switch digitale che prevede l'accensione e lo spegnimento mediante due pulsanti indipendenti, adibiti esclusivamente a questo scopo.

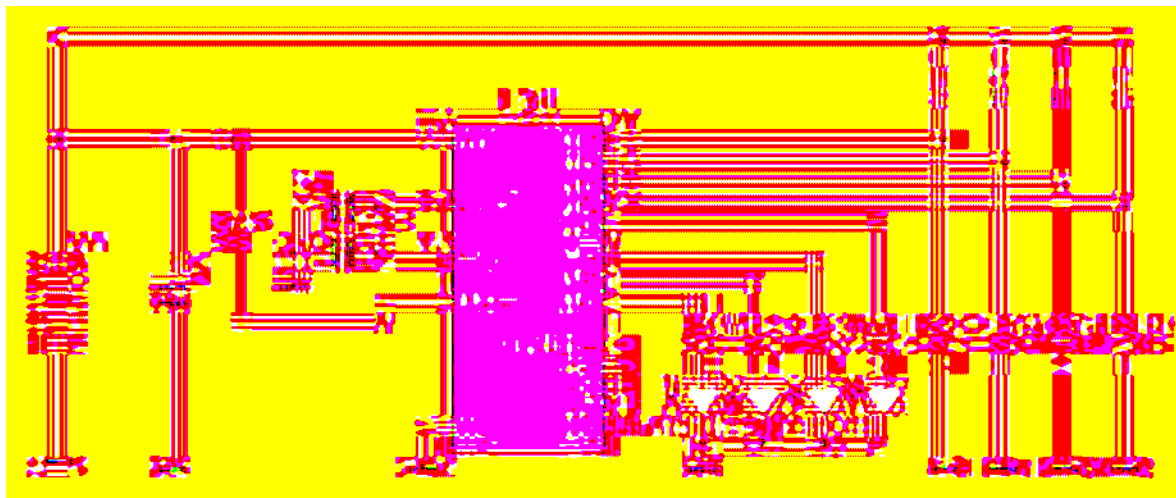



Figura 5 Schema elettrico del progetto con 4 pulsanti e 4 LED

Il solo modo di imparare a programmare un microcontrollore è quello di fare pratica su pratica. Inventate quindi da voi gli esempi e gli esperimenti più disparati, non abbiate paura di sbagliare: nel caso peggiore brucereste qualche componente elettronico di pochi euro. Solo così acquisirete l'esperienza necessaria. Non limitatevi in ogni caso ai soli esempi riportati in questi articoli, ma liberate la fantasia alla ricerca di altri espedienti o addirittura ad altre soluzioni.
Per proporre molti esempi occorrerebbero migliaia di pagine e di illustrazioni, e risulterebbero sempre insufficienti.
Ecco un altro esempio da provare, questa volta un po' più complicato.
Ecco il problema da risolvere:

- Si hanno quattro pulsanti e 4 diodi LED.
- I 4 LED saranno collegati su RB0, RB1, RB2, RB3 (Output).
- I 4 pulsanti (normalmente aperti) saranno collegati su RB4, RB5, RB6, RB7 (Input).
- Inizialmente i 4 LED saranno spenti.
- Se l'utente non preme alcun tasto non succede nulla.
- Se si preme il primo pulsante (su RB4) i quattro LED lampeggeranno per cinque volte (5) alla frequenza di 1 Hz, poi si spegneranno.
- Se si preme il secondo pulsante (su RB5) i quattro LED si accenderanno in sequenza per un secondo (1 secondo il primo, un secondo il secondo, un secondo il terzo e un secondo il quarto).
- Se si preme il terzo pulsante (su RB6) i quattro LED staranno accesi per tre secondi, poi si spegneranno.
- Se si preme infine il quarto pulsante (su RB7) i quattro LED lampeggeranno per cinquanta volte (50) alla frequenza di 100 millisecondi.
- Il software deve essere ciclico ed infinito, cioè una volta terminato un evento, aspetterà il verificarsi del successivo.

Abbiamo complicato veramente le cose: il programma in questione è proprio contorto, richiede una visione completa del problema, ma non preoccupatevi. Cercate di compenetrarvi nell'analisi della questione e vedrete che pian piano troverete anche voi la soluzione. Gli articoli di questa rubrica non vanno letti come un romanzo o un libro giallo. Al contrario essi vanno studiati attentamente. Ma con la passione e la costanza si disporrà sicuramente di una marcia in più.

Bisogna realizzare un prototipo costituito da 4 ingressi e 4 uscite. Si può tranquillamente usare una basetta millefori, per minimizzare le operazioni di realizzazione. Seguiamo lo schema di figura 5.
Scorgiamo subito i quattro Led collegati alla

RB0-1-2-3 e i pulsanti collegati alla RB4-5-6-7. Si notino le resistenze di pull-down che forzano gli ingressi del Pic a livello logico basso per i pulsanti non premuti, mentre in caso contrario gli stessi assumeranno il valore logico alto, proprio perché collegati direttamente all'alimentazione. Le resistenze di limitazione dei diodi Led sono di 220 ohm, che consentono di alimentare i Led stessi con una corrente di (5-2)/220 = 13,6 mA circa, corrente che garantirà una corretta illuminazione.

Andiamo ora ad esaminare il diagramma di flusso proposto in figura 6. L'insieme dei compiti che svolge il microcontrollore ed il metodo usato per risolvere i vari problemi è definito **algoritmo**.

Digitiamo il programma cercando di non commettere errori, in seguito lo commenteremo:

```
program QuattroQuattro

dim k as byte

main:

  portb=0 'Azzera PORTB prima

  'Definisce RB7,RB6,RB5,RB4 come INPUT e
  'RB3,RB2,RB1,RB0 come OUTPUT
  trisb=%11110000  'Definizione PORTB

  while true       'Inizio Ciclo INFINITO
    '-----Se si preme tasto 1----
    if portb.4=1 then
      for k=1 to 5
        portb.0=1 'Led 1 ON
        portb.1=1 'Led 2 ON
        portb.2=1 'Led 3 ON
        portb.3=1 'Led 4 ON
        delay_ms(500)
        portb.0=0 'Led 1 OFF
        portb.1=0 'Led 2 OFF
        portb.2=0 'Led 3 OFF
        portb.3=0 'Led 4 OFF
        delay_ms(500)
      next k
    end if
    '-----Se si preme tasto 2----
    if portb.5=1 then
      portb.0=1 'Led 1 ON
      delay_ms(1000)
      portb.0=0 'Led 1 OFF
      portb.1=1 'Led 2 ON
      delay_ms(1000)
      portb.1=0 'Led 2 OFF
      portb.2=1 'Led 3 ON
      delay_ms(1000)
      portb.2=0 'Led 3 OFF
      portb.3=1 'Led 4 ON
      delay_ms(1000)
      portb.3=0 'Led 4 OFF
    end if
    '-----Se si preme tasto 3----
    if portb.6=1 then
      portb.0=1 'Led 1 ON
      portb.1=1 'Led 2 ON
      portb.2=1 'Led 3 ON
      portb.3=1 'Led 4 ON
      delay_ms(3000)
      portb.0=0 'Led 1 OFF
      portb.1=0 'Led 2 OFF
      portb.2=0 'Led 3 OFF
      portb.3=0 'Led 4 OFF
    end if
    '-----Se si preme tasto 4----
    if portb.7=1 then
      for k=1 to 50
        portb.0=1 'Led 1 ON
        portb.1=1 'Led 2 ON
        portb.2=1 'Led 3 ON
        portb.3=1 'Led 4 ON
        delay_ms(50)
        portb.0=0 'Led 1 OFF
        portb.1=0 'Led 2 OFF
        portb.2=0 'Led 3 OFF
        portb.3=0 'Led 4 OFF
        delay_ms(50)
      next k
    end if
    '------------------------
  wend  'Ritorna al ciclo infinito
end.
```

Figura 6 Flow Chart del progetto con 4 pulsanti e 4 LED

Il listato è molto chiaro. È stato inserito un nuovo comando, o meglio, una nuova struttura: il blocco di codice *while - wend*. Esso consente di migliorare la leggibilità del programma e aggiunge eleganza alla struttura stessa. In pratica, tutto il codice che è racchiuso all'interno di queste due istruzioni, viene eseguito sino a quando la condizione specificata dopo il comando *while* risulta vera. Dal momento che, come condizione, abbiamo messo la parola riservata *true* (vero), il ciclo si ripeterà all'infinito. Il funzionamento è simile all'uso della label con l'istruzione goto;
Qui la label è sostituita dal *while*, mentre il *goto* è sostituito dal wend. Prendete però con le pinze questo paragone: è solo per raffrontare i due metodi, ma in realtà si tratta di comandi completamente diversi.
Con questo esempio abbiamo realizzato un completo sistema automatizzato: un insieme di periferiche in uscita che è costituito dai diodi Led, un insieme di periferiche indipendenti in entrata, costituito dai pulsanti, un sistema di lettura eventi esterni, un blocco di decisione ed un blocco di esecuzione.
Anche i software più complessi (lo vedremo nelle prossime puntate) saranno basati sempre su questi principi basilari che governano i sistemi automatizzati.
A questo punto provate tranquillamente il prototipo creato, e se avete seguito attentamente gli appunti, avendo trascritto correttamente anche il codice, noterete, con grande soddisfazione, che il programma si comporta egregiamente eseguendo ciecamente i comandi impartiti dall'uomo. I comandi da pulsante possono essere ripetuti diverse volte. Il PIC eseguirà il suo lavoro nel migliore dei modi.
Una nota particolare: durante l'esecuzione di un evento dei quattro previsti (parliamo dei vari lampeggi) i tasti non risponderanno ai comandi. Mentre cioè sta verificandosi un lampeggio, anche se si preme uno dei quattro pulsanti, non accadrà nulla.
Questo avviene perché il controllo del software è vincolato all'esecuzione dei comandi in un ciclo "bloccato" e, in teoria, mentre si esegue un processo non ne può avvenire un altro (multitasking).

PROGETTO ILLUMINAZIONE SCALA

Il prossimo progetto è quello che prevede l'illuminazione di un garage o di una scala di un condominio. Esso è realizzato in versione ridotta e a 12 Volt.
Tuttavia è estremamente didattico poiché ripropone tutte le fasi teoriche, pratiche e realizzative che devono essere seguite.
Iniziamo subito dall'analisi del problema da risolvere, e quindi all'ideazione dell'algoritmo.

Analisi del problema

In un scala condominiale solitamente le luci sono spente, per un motivo prettamente economico. In prossimità degli interruttori restano però illuminate delle piccole lampadine che anche al buio danno la possibilità di orientarsi e di trovare subito il pulsante.
Se a questo punto quest'ultimo viene premuto,

si accendono immediatamente le luci della scala, per un periodo di tempo deciso in sede condominiale. Ma contemporaneamente le piccole lucine di orientamento si spengono, essendo ormai del tutto inutile il loro funzionamento. Dopo questo breve tempo di "attività" del circuito, le luci scala si spengono nuovamente e si riaccendono le piccole lampadine di orientamento; il sistema resta quindi in attesa di un nuovo impulso del pulsante. Questo è un classico esempio di un programma a "ciclo infinito".
In una delle prossime puntate proporremo la versione più sofisticata ed "intelligente" del progetto, che prevede anche lo spegnersi delle lampade scala in maniera graduale.

Hardware necessario

Il materiale necessario è ridotto ai minimi termini, almeno in questa versione didattica. Al

Figura 7 Flow Chart del progetto Illuminazione Scala

circuito di pilotaggio vengono aggiunte le seguenti componenti aggiuntive:

- Un pulsante n/a collegato a RB0.
- Un diodo LED collegato a RB1.
- Una lampada di potenza a 12 Volt collegata a RB2 (e pilotata naturalmente da un transistor di potenza).

Algoritmo generale

Le fasi che il microcontrollore dovrà gestire per eseguire il software programmato sono le seguenti:

1. Inizializza ed azzera le porte.
2. Accende piccola lampadina di orientamento e spegne luci scala.
3. Se l'uomo preme il pulsante:
   a. Accende luci scala di potenza
   b. Spegne lampadina di orientamento (rappresentata dal LED)
   c. Mantiene tale stato per 2 minuti
   d. Spegne luci scala di potenza
   e. Accende lampadina di orientamento
4. Ritorna al punto 3 (ciclo infinito).

Rappresentazione mediante Flow Chart

Rappresentiamo in figura 7 lo schema dell'algoritmo in formato grafico, utilizzando i blocchi standard per i diagrammi di flusso.
Lo scopo di questo diagramma è quello di visualizzare schematicamente ed immediatamente il "ragionamento" che seguirà il PIC.

Il listato Basic

Il seguente listato Basic è molto semplice da capire. Eccolo di seguito:

```
program Scala

dim k as byte

main:
  portb=0 'Azzera PORTB

  trisb.0=1 'RB0 in Input
  trisb.1=0 'RB1 in Output
```

```
  trisb.2=0 'RB2 in Output

  portb.1=1 'Accende LED
  portb.2=0 'Spegne lampada

  while true
    if portb.0=1 then  'Se premo pulsante
      portb.2=1 'Accende Lampada
      portb.1=0 'Spegne Led
      '---Attesa---
      for k=1 to 120 'Secondi
        delay_ms(1000) 'Attende un secondo
      next k
      '---------------
      portb.2=0 'Spegne lampada
      portb.1=1 'Riaccende Led
    end if
  wend

end.
```

I commenti del software si spiegano da soli. Si deve approfondire il discorso della temporizzazione dei 120 secondi. Potremmo in teoria usare l'istruzione *delay_ms*(120000) per attendere 120 secondi. Purtroppo tale funzione built-in accetta solo valori in formato *word* (2 byte). Accetterebbe, almeno in teoria, un numero massimo di 65535 millisecondi (65 secondi).
Superiamo allora l'ostacolo facendo ripetere al programma, per 120 volte (for-next), l'attesa di un secondo, e in definitiva otteniamo in maniera precisa i due minuti necessari.
Usando tale metodo si possono effettuare temporizzazioni lunghissime, anche di centinaia di anni…

Schema elettrico

In figura 8 possiamo vedere lo schema elettrico, peraltro molto semplice. Vi sono questa volta due sorgenti di alimentazione: i 5V per alimentare il microcontrollore e i 12V per alimentare la lampada di potenza. Ecco il funzionamento. Se il pulsante (normalmente aperto) non viene premuto, la porta RB0 assumerà valore logico basso in quanto collegata a massa attraverso la resistenza di pull-down da 1 Kohm.
Non appena viene premuto il pulsante, il piedino RB0 assume subito il valore logico alto, poiché avviene il collegamento diretto con i 5 Volt di alimentazione. Il programma si accorgerà subito di questo cambio di stato e, attraverso la nostra routine appositamente creata, si comporterà di conseguenza.
Ricordiamo sempre che all'ingresso devono essere sempre applicati i 5 volt per il livello logico alto e 0 Volt per il livello logico basso. Non bisogna MAI superare questa soglia pena la distruzione del PIC.
Il software, accorgendosi del pulsante premuto, porrà a livello logico alto il piedino RB2, in modo che, attraverso la resistenza di polarizza-

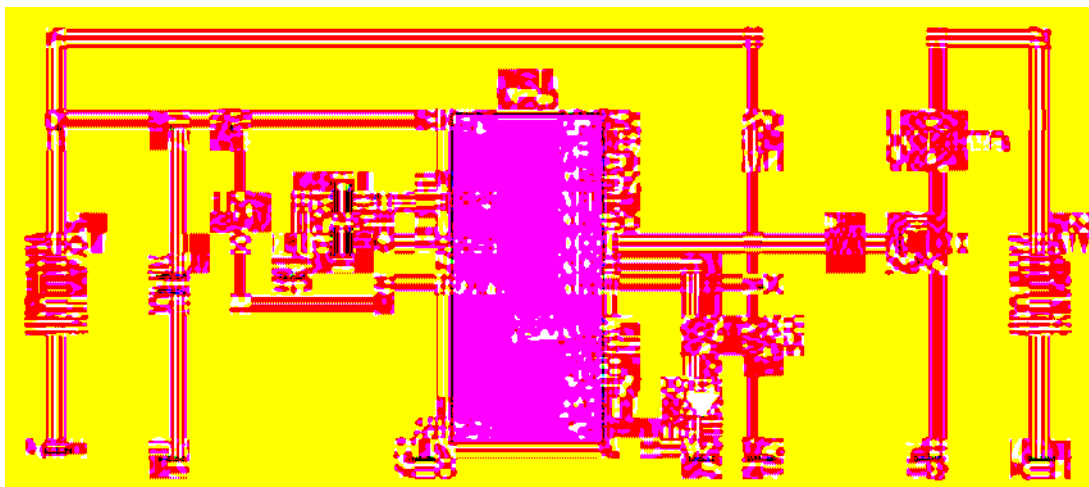



Figura 8 *Schema elettrico del progetto Illuminazione Scala*

zione di base del transistor di potenza BD243, scorra una corrente di circa 20 mA. Tale corrente è più che sufficiente a saturare il transistor (con Beta prossimo a 100) e porlo in conduzione diretta, quindi ad illuminare la lampada.
Occorre dire che il BD243 sopporta bene fino a 6 A di corrente, ma è opportuno dotarlo di apposita aletta di raffreddamento poiché, in ogni caso, scalda parecchio.
Chi volesse può facilmente cambiare la durata di accensione della lampada.
Basta semplicemente sostituire il valore 120 (nel comando *for k=1 to 120*) con altro valore desiderato (ricordiamo però che la variabile k è definita come tipo ***byte***, quindi accetta al massimo un valore di 255).

Figura 9 *L'utilissimo prototipo del Pic Test Board*

Figura 10 *La barra dei LED per testare i programmi*

MIKROBASIC SUL WEB

Trovate il compilatore mikrobasic sul sito http://www.mikroelektronika.co.yu, naturalmente in versione limitata (il limite consiste nella massima estensione del codice generato). Potete scaricarlo liberamente ed installarlo subito sul vostro PC.
Consigliamo anche di scaricare e di stampare il manuale utente che, anche se in lingua inglese, dà moltissime informazioni utili sul suo utilizzo. Per ciò che concerne l'hardware di trasferimento del codice eseguibile (Hex) sul PIC è necessario un programmatore USB, seriale o parallelo connesso al PC.
Sui precedenti numeri trovate anche molti progetti a riguardo.
Per quanto riguarda il software di trasferimento consigliamo l'utilizzo del famoso *IC-Prog*, scaricabile liberamente da Internet e rintracciabile un po' ovunque sui siti.

ESERCITAZIONI

Dal momento che la pratica è la migliore maestra, perché non provare a realizzare "da soli" qualche piccola esercitazione?

Semaforo di un incrocio stradale: provate a realizzare un semaforo che preveda il controllo del traffico in un incrocio stradale. Pertanto i semafori devono essere due: uno per ciascuna strada. Il funzionamento deve essere il seguente:

- Semaforo A rosso, semaforo B verde.
- Semaforo A rosso, semaforo B verde + giallo.
- Semaforo A verde, semaforo B rosso.
- Semaforo A verde + giallo, semaforo B rosso.
- Ripete il ciclo all'infinito.

Usate una temporizzazione sufficiente per gestire i due eventi.

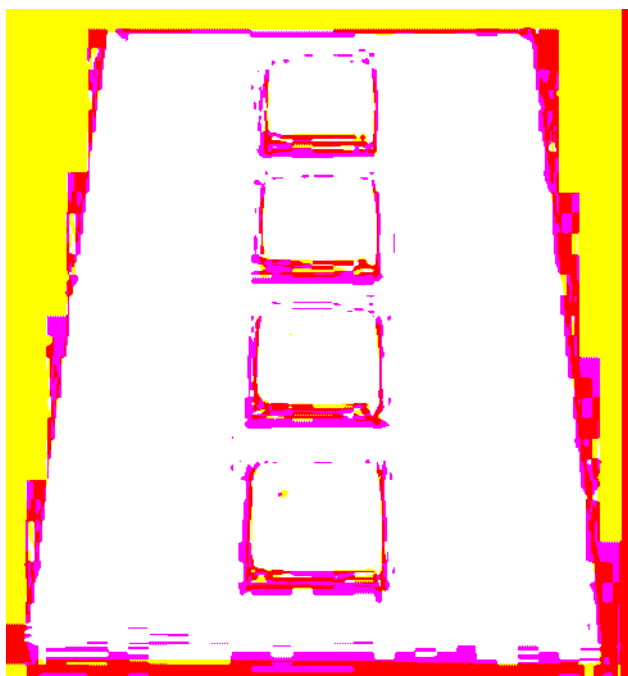



Figura 11 La pulsantiera per testare i programmi

COSA BOLLE IN PENTOLA

Non perdete assolutamente i prossimi numeri della rivista. Se questo articolo, unito al precedente, costituisce solo l'assaggio, il bello deve ancora arrivare.
Nelle parti successive saranno approfondite le seguenti problematiche: utilizzare al meglio le uscite del PIC, utilizzare al meglio gli ingressi del PIC, la teoria del PWM, facciamo musica col microcontrollore, utilizzo dei motori passo-passo, la memorizzazione dei dati, le conversioni analogico digitali, realizzazioni pratiche e divertenti per tutti.

CONCLUSIONI

Il PIC non serve "solamente" per accendere o spegnere led e lampade, come abbiamo visto fino ad ora. In questi primi due articoli abbiamo soltanto mosso i primi passi nella programmazione dei microcontrollori.
Gli esperimenti fin qui condotti hanno avuto il carattere della dimostrazione e della prova, abbiamo trattato la materia con semplicità senza spingerci in ragionamenti troppo tecnici, almeno per il momento.
Nei prossimi numeri saranno trattati, con il dovuto approfondimento, gli argomenti che permetteranno di allargare il proprio bagaglio culturale sulla conoscenza dei PIC, ma soprattutto verranno presi in considerazione gli aspetti pratici di utilizzo che quotidianamente si ha l'esigenza di soddisfare.

Alle prossime puntate.

Parameter	Value
Voltage on VDD with respect to VSS	-0.3 to +7.5V
Voltage on $\overline{MCLR}$ with respect to VSS(1)	-0.3 to +14V
Voltage on RA4 with respect to VSS	-0.3 to +8.5V
Total power dissipation(2)	800 mW
Maximum current out of VSS pin	150 mA
Maximum current into VDD pin	100 mA
Input clamp current, IIK (VI < 0 or VI > VDD)	± 20 mA
Output clamp current, IOK (VO < 0 or VO > VDD)	± 20 mA
Maximum output current sunk by any I/O pin	25 mA
Maximum output current sourced by any I/O pin	25 mA
Maximum current sunk by PORTA	80 mA
Maximum current sourced by PORTA	50 mA
Maximum current sunk by PORTB	150 mA
Maximum current sourced by PORTB	100 mA

Figura 12 Valori massimi elettrici sopportati dal PIC (dal Datasheet della Microchip)

SE L'ELETTRONICA È LA TUA PASSIONE GLI INDIRIZZI DEI GBC *Store* DI TUTTA ITALIA TI SARANNO SICURAMENTE UTILI...

Filtro di Kalman

Il filtro di Kalman appartiene alla categoria dei filtri che operano nel dominio del tempo e rappresenta una metodologia di calcolo che fornisce la stima di un segnale affetto da rumore

Dal 1960, anno in cui R.E. *Kalman* ha pubblicato i suoi risultati, questo algoritmo è stato oggetto di intense ricerche e applicazioni, in particolare nel campo della navigazione aerea. Ad esempio, la NASA utilizzò un sistema di telemetria, che faceva uso di questo filtro (all'epoca delle missioni Apollo), per calcolare la distanza tra il modulo LEM e la superficie del nostro satellite durante le fasi di allunaggio.

Il filtro di Kalman comprende un insieme di equazioni matematiche, implementabili mediante un semplice programma di calcolo, che consentono di stimare una grandezza (quale ad esempio la posizione o la velocità di un aeromobile), anche in presenza di notevoli perturbazioni sulle misure.

Il calcolo si basa sulle conoscenze statistiche del rumore e non necessita di una precisa modellazione del sistema da controllare.

Un'area tipica in cui il filtro di Kalman trova applicazione è rappresentata dal sistema inerziale giroscopico di un aeroplano, ma si utilizza vantaggiosamente anche per il filtraggio di immagini video affette da disturbi (vedere figura 1).

Figura 1 Immagine prima e dopo il filtraggio

APPLICAZIONE PER UN CONTROLLO AUTOMATICO DI GUADAGNO

In un ricevitore supereterodina gestito a microcontrollore, la procedura software di gestione dell'AGC svolge il compito di effettuare il loop di regolazione del guadagno della catena IF mediante osservazione del valore di picco del segnale rivelato e, di conseguenza, pilotaggio degli stadi di attenuazione, allo scopo di ottenere un valore di segnale quanto più prossimo ad una costante (figura 2).

Una fase essenziale di tale procedimento è quindi rappresentata dal processo di **stima** del segnale rivelato in banda base.
Tale stima deve essere la più accurata possibile, in modo da ridurre l'influsso del rumore, e sufficientemente rapida per poter "seguire" fedelmente qualsiasi variazione del segnale in ricezione.

di Agostino Rolando
a.rolando@farelettronica.com

e sue applicazioni

Figura 2 Catena di regolazione AGC

Una modalità di stima frequentemente utilizzata prevede di acquisire un certo numero N di campioni di segnale alla volta; questi vengono memorizzati, ad esempio in una memoria FIFO e, infine, su di essi viene attuata una media aritmetica.

In questo modo, tuttavia, a seguito dell'operazione di media il valore stimato risente inevitabilmente di un certo ritardo (proporzionale ad N) per cui, se da un lato si ottiene una buona stabilità nella regolazione del guadagno, dall'altro può tuttavia essere difficoltoso "seguire" un segnale di ingresso che manifesti rapide variazioni.

All'estremo opposto, se per la regolazione dell'AGC si utilizzasse solo il campione corrente del segnale in ingresso, la stima che se ne avrebbe sarebbe sì velocizzata al massimo, ma soffrirebbe di eccessiva instabilità, dando origine a un'inefficace gestione del loop di controllo.

Nel seguito viene descritta un'implementazione di stima basata sul filtro di Kalman. Tale tecnica coniuga i vantaggi del metodo della

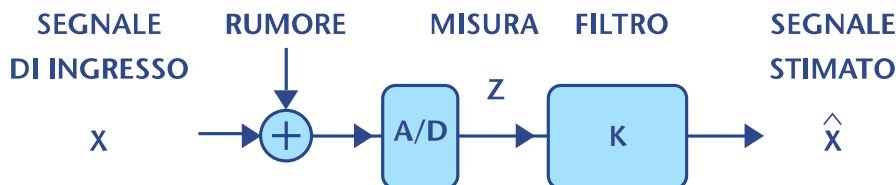



Figura 3 Modellazione del sistema

media aritmetica con la rapidità di utilizzo di un solo campione del segnale di ingresso.

DESCRIZIONE DELL'ALGORITMO

Il circuito sotto esame contiene un convertitore A/D che mette a disposizione,in ingresso al blocco di AGC,dei campioni di tensione aventi un certo valor medio (ad esempio 4 volt) e una variazione, ad esempio, di 0.3 volt intorno a questo valore (figura 3).
Il segnale di ingresso può quindi essere convenientemente modellato con una variabile aleatoria X, identificata da un valor medio e da una varianza (come da figura 4).

Prendiamo come riferimento la figura 5 (ottenuta con una simulazione MATLAB), dove il segnale X è rappresentato dalla traccia in rosso e i valori che vengono effettivamente misurati, denominati Z, sono indicati dalla traccia in blu. Esiste una certa differenza tra i due segnali, dovuta alla perturbazione che definiremo "rumore di misura".

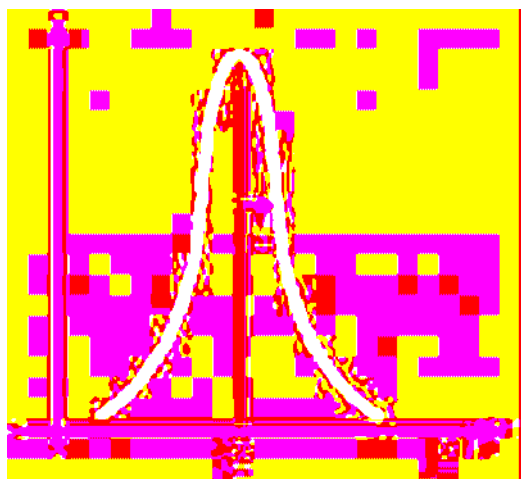



Figura 4 variabile aleatoria X e sua deviazione standard

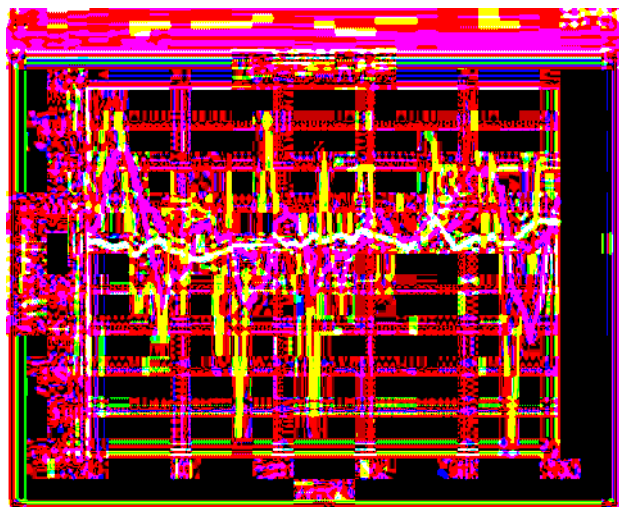



Figura 5 Prestazioni del filtro

Tale rumore è comprensivo di tutte le cause di disturbo, come ad esempio le interferenze dovute a segnali su canali adiacenti o lo stesso errore di quantizzazione del convertitore A/D.

L'algoritmo di filtraggio predittivo di Kalman, in sostanza, realizza il concetto di una media che si "evolve" nel tempo; per cui non è necessario, per ogni nuovo campione che arriva, rifare tutta la serie di conti già effettuata sui campioni precedenti (così come avviene per la media tradizionale).
Infatti, questo tipo di stima ha una caratteristica **ricorsiva**; in altri termini, quello che si ottiene è un filtro la cui funzione di trasferimento (K, in figura 3), ovvero il suo guadagno, non è costante, ma si "aggiorna" in continuazione, a mano che si acquisiscono i nuovi campioni.
Tale guadagno, in pratica, conserva in sè l'informazione sulla storia passata, cioè sui campioni precedenti del segnale da stimare.
Inoltre, un'altra caratteristica, tipica di questo algoritmo consiste nel poter fornire una valutazione in real-time dell'errore che si commette sulla stima medesima.
Come abbiamo già detto, trattandosi di variabili aleatorie, la stima sarà convenientemente rappresentata da un valor medio e da una varianza, e la varianza rappresenterà l'errore commesso nella valutazione.

Una caratteristica peculiare del filtro di Kalman, che ne costituisce la forza, è la seguente: con il procedere degli step di acquisizione, la varianza sulla stima tende a ridursi sempre più; in altri termini, il filtro, dopo un certo numero di passi... "**converge**".
Inizialmente, il valore stimato sarà inevitabilmente affetto da un certo errore, la cui entità

può essere anche non trascurabile, per cui possiamo figurarci che il valore stimato sarà centrato su una "campana" gaussiana piuttosto ampia. A mano che arrivano i nuovi campioni di ingresso, la campana si stringe sempre più. Di conseguenza, l'errore si riduce progressivamente e la nostra conoscenza del segnale si perfeziona (vedere figura 6).
In altri termini, la gaussiana tende idealmente a un impulso di *Dirac*, il che significa che il valore stimato tende sempre più ad avvicinarsi al valore reale. La stima si compone di due contributi, denominati **predizione** e **innovazione**.

Nel seguito vengono esposti i risultati della simulazione del filtro di Kalman realizzata in ambiente *MATLAB*.

RISULTATI DELLA SIMULAZIONE

L'efficacia del filtro di Kalman è evidenziata dal grafico riportato in figura 5.
In rosso sono rappresentati i valori (ideali, cioè senza rumore aggiunto) dei campioni di partenza, in blu i valori misurati (affetti da rumore) mentre in verde sono evidenziati i valori stimati con il filtraggio.

Possiamo notare la caratteristica **adattiva** del filtro, che si esprime nel fatto che l'uscita del filtro riesce a "seguire" efficacemente le variazioni dell'ingresso, pur in presenza di un forte rumore che si sovrappone al segnale.
Per confronto, la traccia gialla sullo stesso grafico rappresenta la stima che si otterrebbe con il metodo della **media aritmetica**, effettuata su *N=quattro* campioni consecutivi di segnale.

Il grafico successivo, in figura 7, mostra invece la varianza dell'errore sulla stima, in altri termini come si evolve l'errore commesso dal filtro.
Si può notare come questo tenda al valore limite di 0.1 in 10-4 dopo 20 step. In altre parole, il filtro **converge** dopo una ventina di passi; ciò significa che, dopo aver acquisito venti campioni, siamo garantiti di avere questa entità di errore sulla stima.

BANDA PASSANTE

Osserviamo che la rapidità di convergenza del filtro, che è strettamente connessa alla sua banda passante, è funzione del clock con cui campioniamo il segnale. Nella procedura MATLAB qui esposta, il periodo di clock è rappresentato dal parametro "dt", che viene posto a valore 1.
Se invece ponessimo, ad esempio, *dt = 0.1*, significherebbe che il nostro clock di campionamento sarebbe 10 volte più veloce.
Di conseguenza, sarebbe sufficiente un periodo di tempo dieci volte inferiore per raggiungere la convergenza, quindi la banda passante del filtro ne sarebbe proporzionalmente accresciuta.
Il valore 0.1 in 10-4 rappresenta il **valore limite di stazionarietà** dell'errore. Esso è tanto più piccolo quanto più stabile è il segnale di ingresso. A titolo di esempio, se considerassimo il caso estremo di un segnale di ingresso perfettamente costante nel tempo, tale valor limite si ridurrebbe praticamente a zero.

VANTAGGI

Uno dei vantaggi del filtro di Kalman consiste nel fatto che una buona parte dei calcoli (quelli per esempio sulla varianza) può essere eseguita in momenti distinti da quelli di acquisizione, con conseguente velocizzazione della computazione complessiva.

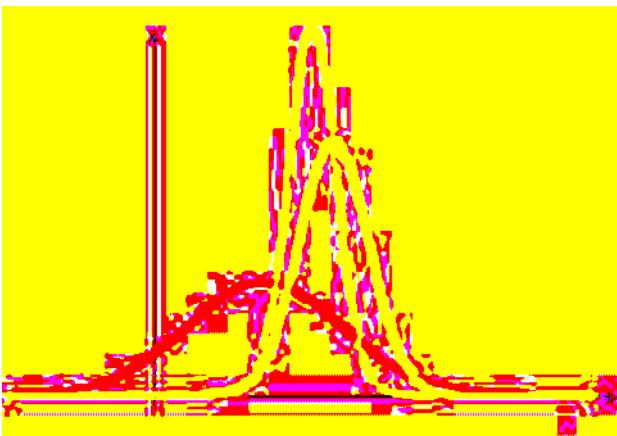



Figura 6 *Processo di convergenza del filtro*

CONCLUSIONI

Con l'evolversi della tecnologia, il Filtro di Kalman si presenta come una valida alternativa ai classici algoritmi di filtraggio digitali.
Infatti, fino a pochi anni fa, l'unico modo per mettere in atto questa metodologia consisteva nell'utilizzo di programmi software, ottimizzati per microcontrollori o DSP, i quali tuttavia presentavano una sostanziale limitazione nella banda di segnale gestibile, limitazione dovuta in gran parte all'elevato tempo di elaborazione che questo tipo di filtro, per sua natura, richiede.
Per contro, oggigiorno il mercato mette a disposizione dispositivi programmabili FPGA (ad esempio la famiglia *Virtex IV* della *Xilinx Corp.*), che ben si prestano a tale tipo di implementazione.
L'incremento delle prestazioni che questi dispositivi hanno raggiunto negli ultimi anni, sia in termini di capacità che di frequenza di clock di sistema e di funzioni matematiche embedded (come i blocchi moltiplicatori a 18 bit), rende oggi attuabili e convenienti gli algoritmi di filtraggio come quello di Kalman anche per applicazioni a frequenze video, ad esempio per il filtraggio di immagini in movimento.
Un esempio è dato dalla figura 8, in cui vengono riportati due frame di un'immagine video.
Nel caso a) viene applicato un filtro di tipo "mediano", che riduce il rumore effettuando una media su alcuni pixel.
Nel caso b), il frame viene sottoposto a filtro di Kalman in una versione perfezionata con "motion detection". Sempre nella figura 8 b) vediamo inoltre il risultato che si otterrebbe con l'applicazione di un filtro di tipo non adattivo.

Figura 7 Andamento dell'errore sulla stima

PROCEDURA DI SIMULAZIONE

Nel seguito viene esposta la procedura di simulazione utilizzata.
Essa genera i campioni di segnale X, in maniera random; a questi campioni viene quindi sommato un opportuno "rumore" aleatorio al fine di simulare un segnale con variazioni realistiche. Infine, sulla misura Z

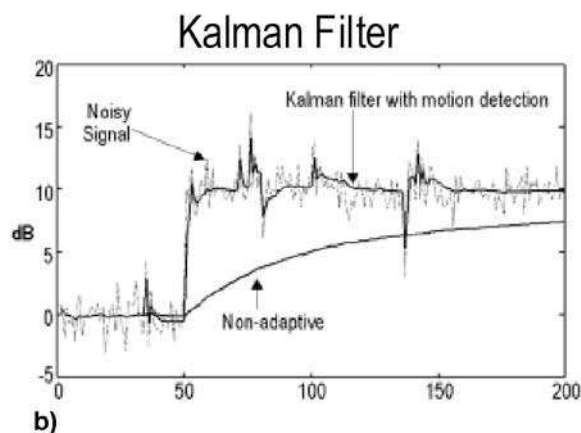
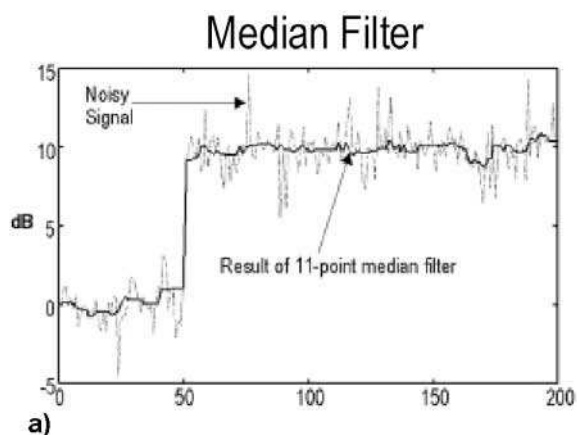



Figura 8 Confronto tra filtro mediano e di Kalman

Figura 9 *Schema a blocchi del filtro*

degli stessi viene effettuata la stima, "ottima" secondo Kalman.
Ci riferiamo sempre alla figura 3.

In dettaglio, il procedimento si svolge secondo la seguente modalità:

- Per poter simulare con una procedura software la generazione dei campioni X, si assume che il segnale sia una variabile aleatoria, caratterizzata da un certo "rumore di processo" gaussiano, bianco, con valor medio = 0 e deviazione standard "procnoise", che si sovrappone al valore costante di riferimento X = 4.
- La misura viene a sua volta identificata con la grandezza aleatoria Z, la quale sarà perturbata da rumore gaussiano, bianco, a valor medio=0 e avente una certa deviazione standard chiamata "measnoise".

Nella realtà, Z rappresenta il valore che viene effettivamente campionato e memorizzato in una memoria fifo.
L'algoritmo prevede una fase di inizializzazione e un successivo loop, ripetuto per tutti i campioni che vengono acquisiti.

Gli step sono i seguenti (vedere schema a blocchi in figura 9):

- Inizializzazione dello stato, dove con "stato" si intendono i campioni ideali.
- Inizializzazione della stima dello stato.
- Inizializzazione della varianza della stima.
- Inizializzazione varianza dell'errore di misura.
- Inizializzazione varianza del processo.
- Generazione di un nuovo valore di X.
- Misura.
- Calcolo della predizione dello stato.
- Calcolo della varianza sulla predizione dello stato.
- Calcolo del termine di Innovazione.
- Calcolo Della varianza dell'Innovazione.
- Calcolo del "guadagno" K del filtro.
- Calcolo della stima a posteriori dello stato.
- Calcolo della varianza dell'errore sulla stima a posteriori.

I valori così ottenuti (valore vero X,valore misurato Z, valore stimato X^) vengono ad ogni step "salvati" in 3 rispettivi vettori, per essere successivamente plottati al termine dell'elaborazione.
Così facendo, è possibile valutare a posteriori l'efficienza del filtro.

Segue l'algoritmo implementato per MATLAB:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulazione filtro di Kalman
% durata = lunghezza della simulazione
% dt     = step
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function filtroK
  durata=50;
  dt = 1;

  procnoise  = 0.001;    % dev. std del "rumore di processo"
  measnoise  = 0.01;     % dev. std del "rumore di misura"

  x = 4;                 % inizializz. stato
  xhat = x;              % inizializz. stima dello stato
  Q = procnoise^2        % varianza del rumore di processo
  R = measnoise^2;       % varianza dell'errore di misura
  P = 2;                 % inizializz. varianza sulla stima (dont'care)

% vettori per plottaggio dati
  pos       = [];        % valori veri, x
  poshat    = [];         % valori stimati, xhat
  posmeas   = [];         % valori misurati, z
  varst     = [];         % varianza della stima
  media     = [];         % medie dei campioni
  Counter = 0;            % iniz. contatore del loop
  fifo1 = 4;              % iniz. fifo
  fifo2 = 4;
  fifo3 = 4;
  fifo4 = 4;

% Loop principale
  for t = 0 : dt: durata,

        Counter = Counter + 1;

    % Simula il processo, cioè genera  valori reali di x
    % mediante applicazione di un rumore random di processo
        ProcessNoise = procnoise * randn;
        x = x + ProcessNoise;

    % Simula le misure, cioè genera valori di z
    % mediante applicazione di un rumore random di misura
    % sui valori x
        MeasNoise = measnoise * randn;
        z = x + MeasNoise;
```

```
    % shifta la fifo di una posizione
        fifo1 = fifo2;
        fifo2 = fifo3;
        fifo3 = fifo4;

    % inserisce il campione corrente
        fifo4 = z;

    % ogni 4 campioni, calcola la media aritmetica
        med = (fifo1 + fifo2 + fifo3 +fifo4) /4;

    % Calcola la varianza della stima a priori
        P = P + Q;

    % Calcola il termine di Innovazione
        Inn = z - xhat;

    % Calcola la varianza dell'Innovazione
        S = P + R;

    % Calcola il guadagno K del filtro
        K = P / S;

    % Calcola la stima "a posteriori"
        xhat = xhat + K * Inn;

    % Varianza dell'errore sulla stima "a posteriori"
        P = P * (1 - K)*(1 - K)  + K * R * K;

    % Salva i valori in opportuni vettori, per successivo plottaggio
        pos     = [ pos; x ];   % campione vero
        posmeas = [ posmeas; z ];   % valore misurato
        poshat  = [ poshat; xhat ];   % valore stimato
        varst = [ varst; P ];   % Varianza dell'errore sulla stima
        media = [ media; med ];   % Valore medio
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Plottato dei risultati
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        t = 0 : dt : durata;
        t = t';
    % ROSSO = VALORE VERO DI X
    % BLU   = MISURA
    % VERDE = VALORE STIMATO
```

```
% GIALLO = MEDIA su 4 campioni consec.

     plot(t,pos,'r',t,poshat,'g',t,posmeas,'b',t,media,'y');
     grid;
     xlabel('Tempo ');
     ylabel('Ampiezza (Volt))');
     title('Prestazioni K Filter');

% Plottato della sola varianza dell'errore sulla stima
%      plot(t,varst,'y');
%      grid;
%      xlabel('Tempo ');
%      ylabel('Errore sulla stima');
%      title('Prestazioni K Filter');
```

NOTE BIOGRAFICHE

Il nome di R.E.Kalman è legato alla omonima metodologia matematica per la stima di segnali affetti da rumore, da lui concepita nel 1960. Rudolf Emil Kalman nacque a Budapest, Ungheria, il 19 maggio del 1930.
Trasferitosi negli USA, si laureò in ingegneria elettronica al *Massachusetts Institute of Technology* nel 1953 e conseguì il dottorato alla *Columbia University* nel 1957.
In seguito, tra gli anni 1958 e1964, svolse l'attività di ricercatore matematico all'istituto *RIAS* di *Baltimora*.
Professore alla *Stanford University* tra gli anni 1964 e 1971, divenne in seguito direttore del *Centro di Studi Teorici Matematici* all'*Università della Florida*.

Successivamente, tenne la cattedra di matematica all'*Istituto Svizzero Federale di Tecnologia*, con sede a *Zurigo*.
Al suo attivo conta numerosi riconoscimenti, tra cui la *Medaglia d'Onore dell'Istituto IEEE* (1974), il *Premio Kyoto in Alta Tecnologia*, conferitogli dalla *Fondazione Inamori* nel 1985, lo *Steele Prize* della *American Mathematical Society* (1987) e il *Bellman Prize* (1997).

Figura 10 *Rudolf Emil Kalman*

È membro dell'*Accademia Nazionale delle Scienze*, dell'*Accademia Nazionale degli Ingegneri* e dell'*Accademia di Arti e Scienze* (USA).

BIBLIOGRAFIA

Peter S. Maybeck: **Stochastic Models, Estimation and Control, Vol.1**, Academic Press,1979 (http://www.cs.unc.edu/~welch/media/pdf/maybeck_ch1.pdf).

R. G. Brown and P. Y. C. Hwang, **Introduction to Random Signals and Applied Kalman Filtering**, 3rd ed., John Wiley and Sons, New York, 1997.

S. Haykin, **Adaptive Filter Theory**, 3rd Ed., Prentice Hall, New Jersey, 1996.

J. C. Braileam, R.P. Kleihorst, S. Efstratiadis, A. K. Katsaggelos, and R.L. Lagendijk, **"Noise reduction filters for dynamic image sequences: A review"**, Proceedings of The IEEE, vol. 83, no. 9, September 1995.

I vantaggi a te riservati

- ✔ **Risparmierai** ben 15,50 euro sul prezzo di copertina
- ✔ Avrai la garanzia del **prezzo bloccato** per un anno
- ✔ La rivista ti sarà **recapitata comodamente** a casa
- ✔ Compreso con l'abbonamento (o il rinnovo) riceverai un **buono sconto del 20%** per un tuo prossimo acquisto sul sito www.farelettronica.com e, insieme alla merce ordinata, ti sarà recapitato l'esclusivo portapenne di Fare Elettronica

INWARE EDIZIONI

Si approfitto della fantastica occasione e mi abbono a 11 numeri di Fare Elettronica e pagherò solo € 45,00

DATI PERSONALI

Nome ....

Cognome ....

Via .... n° ....

Cap .... Città .... Prov ....

Tel .... Fax ....

Email ....

Ragione Sociale ....

P.Iva .... ❍ Ricevuta

Privacy. Ai sensi del Decr. Lgs. 196/2003 la informiamo che i dati trasmessi verranno impiegati coi principali scopi di indagini di mercato e nelle modalità previste dallo stesso, prevalentemente con mezzi informatici. Il conferimento, di norma facoltativo, è obbligatorio per permettere il rapporto commerciale. È in ogni caso fatto diritto dell'interessato esercitare i propri diritti, nei modi previsti dal "Titolo II art. 7" della legge sopra citata, scrivendo a Inware Edizioni Via Cadorna 27 – 20032 Cormano o trmite email a info@inwaredizioni.it

ABBONATI SUBITO

Compila, ritaglia e spedisci via fax questo coupon allo **02-66508225**

Spedisci questa pagina in busta chiusa a:
INWARE Edizioni
Via Cadorna, 27/31 - 20032 Cormano (MI)

Chiamaci allo **02-66504794**

Abbonati on-line sul sito
www.farelettronica.com/abbonamento

MODALITÀ DI PAGAMENTO

❍ **CARTA DI CREDITO**

❑ American Express ❑ Visa ❑ Mastercard

Titolare ....

n° .... scad ....

❍ **VERSAMENTO SUL CCP N 22790232**

allegare la ricevuta del versamento intestato a Inware Srl indicando nella causale "Abbonamento Fare Elettronica"

❍ **BONIFICO BANCARIO**

Appoggiarlo su: **Poste Italiane - CIN: Z - ABI: 07601 CAB: 01600 - C/C: 000022790232** intestato ad **Inware srl**

❍ **ALLEGO UN ASSEGNO**

intestato a Inware Srl

Firma ....

Alimentatori switching:

Questa puntata, di taglio più teorico, servirà a fornire le prime informazioni necessarie per la realizzazione dei trasformatori per gli switching. Descriverò di seguito alcuni procedimenti per poter misurare le induttanze, pur non avendo a disposizione un misuratore professionale. Infine introdurrò due nuove tipologie che saranno descritte in maggior dettaglio nella prossima puntata, con l'utilizzo di un nuovo circuito integrato switching: l'SG3525.

IL TRASFORMATORE

Abbiamo visto nella puntata precedente, che

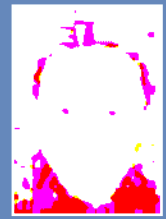
> una corrente elettrica che scorre in un conduttore genera sempre un campo magnetico nello spazio intorno al conduttore. L'intensità del campo magnetico (H) generato è proporzionale alla corrente.

Ricordiamo anche che un materiale ferromagnetico presenta la proprietà di allineare le proprie "isole" magnetiche (cioè il campo magnetico elettronico interno) concordemente al campo magnetico esterno, aumentandone per così dire l'effetto.
Supponiamo ora di "immergere" in un campo magnetico statico (cioè che non varia nel tempo, come ad esempio quello generato da un magnete permanente), una singola spira di un conduttore, collegata ad un sensibile multimetro.
Muoviamo ora lentamente la spira nel campo magnetico, ad esempio avvicinandola o allontanandola dal magnete: il multimetro misurerà una tensione! Tale tensione va a zero se ci fermiamo, mentre aumenterà se il movimento è più veloce. Inoltre la tensione è presente solo se durante il movimento, la spira viene orientata opportunamente rispetto al campo magnetico. Il semplice esperimento appena fatto dimostra la **legge dell'induzione magnetica**, cioè:

> una qualunque variazione di flusso magnetico, induce in un qualunque conduttore (ad esempio una spira) una tensione che dipende proporzionalmente dalla velocità di variazione del flusso.

Tale tensione ha segno tale da opporsi alla variazione di flusso, come è intuitivo: se così non fosse, la tensione indotta aumenterebbe a sua volta il flusso, che aumenterebbe ancora la tensione indotta e così via.
La legge dell'induzione magnetica viene chiamata anche **Legge di Faraday**, oppure Legge di Faraday-Neumann-Lenz, ovviamente dai nomi degli scopritori. La formula 1 ne riporta l'espressione analitica.

Formula n.1
$V = - \Delta\Phi / \Delta t$
V in Volt, Δt in secondi, $\Delta\Phi$ in Weber

di Romano Bernarducci
r.bernarducci@farelettronica.com

I trasformatori per gli switching

La lettera greca phi maiuscola Φ indica il flusso totale, ed è uguale semplicemente alla densità di flusso B, introdotta in FE 238, moltiplicata per l'area di interesse. Dalle formule della puntata precedente, risulta quindi che anche il flusso totale Φ dipende dall'intensità del campo H, e quindi dal numero delle spire N dell'avvolgimento e dalla corrente che vi scorre. L'unità di misura **Weber**, abbreviata normalmente con **Wb**, viene utilizzata per semplicità al posto della complessa espressione seguente:

$$1 \text{ Wb} = 1 \text{ kg} * \text{m}^2 / (\text{A} * \text{s}^2)$$

Ne consegue quindi, che la densità di flusso B si può esprimere anche in termini di Wb / m^2, oltre alle già note unità di misura Tesla e Gauss (tanto per complicare di più le idee!).
La Legge di Faraday è stata applicata ed è alla base del funzionamento di qualsiasi macchina elettrica, come:

- Motori elettrici
- Generatori (alternatori e dinamo)

Figura 1 Il primario del trasformatore

- Azionamenti lineari (solenoidi)
- Trasformatori
- Trasduttori segnale elettrico → pressione acustica (altoparlanti)
- Trasduttori pressione acustica → segnale elettrico (microfoni)
- Trasduttori di posizione
- Azionamenti di precisione (testine hard-disk)

e la lista potrebbe continuare a lungo.
Consideriamo il dispositivo di nostro interesse, cioè il trasformatore. **Un trasformatore si differenzia da una semplice induttanza per il fatto di avere almeno due avvolgimenti isolati, denominati primario e secondario**.
In genere i due avvolgimenti sono avvolti su di un opportuno nucleo ferromagnetico, vedremo più avanti perchè. In figura 1 è riportato il solo primario di un trasformatore avvolto su un nucleo toroidale, di materiale ferromagnetico.

Supponiamo che il primario sia un avvolgimento costituito da 4 spire di filo di rame come in figura. Collegando un generatore sinusoidale all'avvolgimento, verrà creato un campo magnetico alternato H, e quindi un flusso magnetico alternato che percorrerà in una direzione e nell'altra il toroide, con la stessa frequenza del generatore.
Per motivi che non starò qui a spiegare, **praticamente tutto il flusso magnetico generato resta confinato all'interno della ferrite, e si espande solo in minima parte nell'aria circostante**. Questo effetto dipende dalla permeabilità magnetica del materiale, che è molto più alta di quella dell'aria. Avvolgiamo ora un secondario di 2 spire sullo stesso toroide, come in figura 2.

La variazione di flusso all'interno del toroide, generata dalla tensione e corrente variabile nel primario, si accoppia (spesso si usa il termine “si concatena”) con l'avvolgimento secondario. Ai capi di questo, per la Legge di Faraday, sarà presente una tensione data proprio da:

$$Vs = - \Delta\Phi / \Delta t$$

In pratica, è come se il flusso magnetico facesse da “trasportatore di energia” dal primario al secondario.
La Legge di Faraday, applicata al trasformatore, si può manipolare con facili passaggi, in modo da ottenere delle formule di progetto che possiamo riassumere nelle seguenti:

- La tensione sul secondario Vs è pari a quella sul primario Vp, moltiplicata per il rapporto Ns / Np

  Vs = Vp * (Ns / Np) o, equivalentemente:
  Vs / Ns = Vp / Np

- Collegando un carico Rs sul secondario, la corrente Is che vi scorre è pari alla corrente che scorre nel primario Ip, divisa per il rapporto Ns / Np

  Is = Ip / (Ns / Np) cioè:
  Is * Ns = Ip * Np

- Dalle precedenti, applicando la legge di Ohm, la resistenza fittizia Rp vista dal primario, dovuta al carico Rs sul secondario, è pari a:

  $$Rp = Rs / (Ns / Np)^2$$

Per finire, un'interessante proprietà dei trasformatori è la seguente:

in via teorica, il trasformatore è una macchina reversibile, cioè si può collegare un generatore al secondario e prelevare la tensione/corrente trasformata sul primario, invertendo le stesse formule appena citate.

Figura 2 Il trasformatore completo

PERDITE NEI MATERIALI MAGNETICI: PERDITE PER ISTERESI

Supponiamo di utilizzare un materiale ferromagnetico, e precisamente una ferrite, per realizzare un'induttanza per uno switching di tipo buck. Sappiamo che, in generale, per motivi di costo ed efficienza, conviene far lavorare lo switching di tipo buck in modalità CONTINUA, quindi l'induttanza sarà soggetta ad una corrente alternata a rampa (ascendente e discendente) ΔI, con sovrapposta una notevole corrente continua indicata con Im in figura 3.

Cosa succede nel grafico B-H?
Sappiamo che il campo magnetico H (ascissa del grafico B-H), dipende strettamente dalla corrente che scorre nell'avvolgimento. È lecito quindi supporre che anche il campo magnetico H sarà composto da un valore “medio” Hm, generato dalla corrente conti-

Figura 3 Corrente nell'induttanza per tipologia buck

nua Im, con sovrapposto un più piccolo campo ΔH in alternata, generato dalle rampe salita-discesa della corrente. Il grafico esemplificativo è visibile in figura 4.

In pratica, durante un ciclo completo di corrente (rampa ascendente + rampa discendente), nel grafico B-H verrà comunque percorso un ciclo di isteresi. L'area di questo piccolo ciclo rappresenta un'energia, che è quella necessaria ad ogni ciclo per allineare da una parte e dall'altra i campi magnetici interni al materiale. **In definitiva, è una perdita di energia**. Si ottiene quindi, per le perdite magnetiche dovute all'isteresi, la generica formula:

$$P_{DISS,ISTERESI} = f_{SW} * Area(B\text{-}H)$$

Il calcolo diretto è abbastanza complicato, in quanto dipende dal tipo di materiale ferromagnetico, dalla temperatura e dalle dimensioni del nucleo, per cui si preferisce utilizzare apposite tabelle fornite da tutti i produttori di ferriti.

DUE NUOVE TIPOLOGIE, ANZI UNA

Accenniamo ora brevemente a due nuove tipologie di switching che utilizzano il trasformatore e che approfondiremo nella prossima puntata con la realizzazione di un circuito. La prima è la cosiddetta **"forward"**, il cui schema esemplificativo è riportato in figura 5.

Per la precisione, tale configurazione viene definita "forward" **single-ended** in quanto il primario del trasformatore, che è quello collegato al (+) di Vi e all'interruttore 1, viene fatta scorrere corrente sempre e solo in una direzione. Il secondo avvolgimento collegato al diodo D3 ha lo scopo di "resettare" il nucleo, per motivi che vedremo meglio nella prossima puntata. Come si vede, tutti gli avvolgimenti sono in fase, cioè hanno il pallino che indica l'inizio dell'avvolgimento, collegato concordemente: quando si chiude l'interruttore 1, scorre corrente nel primario e **contemporaneamente** scorre corrente nel secondario, secondo la formula già presentata:

$$Is = Ip / (Ns / Np)$$

Una variante di questa tipologia è la cosiddetta "forward" **push-pull**, il cui schema di principio è riportato in figura 6.

A differenza della precedente tipologia, sia il primario che il secondario sono divisi in due avvolgimenti identici.
L'eccitazione del primario viene effettuata alternativamente tra il punto centrale del primario collegato al (+) di Vi, e uno dei due avvolgimenti: in pratica ad ogni semi-ciclo, **il flusso magnetico che scorre nel nucleo cambia direzione**. Ovviamente questo richiede un circuito di controllo con due uscite di pilotaggio, sfasate tra loro di 180°. In figura 7 è riportato il funzionamento alternato della tipologia push-pull.

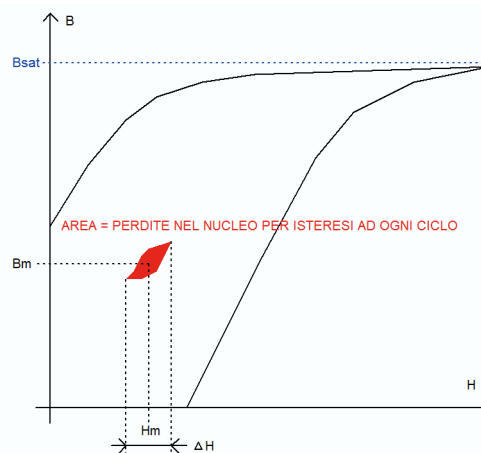



Figura 4 Corrente nell'induttanza per tipologia buck

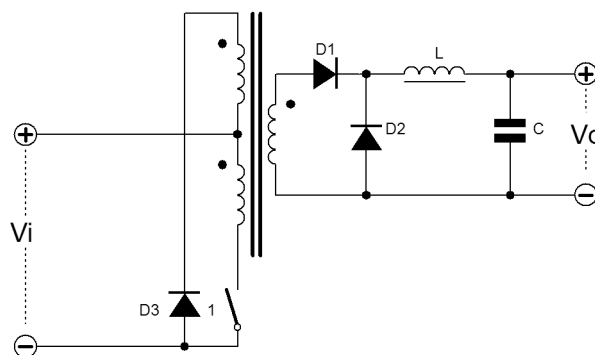



Figura 5 Tipologia a trasformatore "forward" single-ended

Nel prossimo numero presenterò una descrizione più approfondita del funzionamento e delle differenze tra le due tipologie "forward".

INDUTTANZE: MISURAZIONI

Sappiamo che le principali caratteristiche di un'induttanza sono essenzialmente due: il valore dell'induttanza in Henry e la sua resistenza serie in Ohm. Nel caso siano incogniti, come determinare questi parametri? Chi di voi è il fortunato possessore di un ponte LCR, potrà effettuare la misura diretta.
Tuttavia questo strumento non è molto comune tra gli appassionati, ragion per cui vado a presentarvi un paio di metodi molto semplici, che richiedono solamente un minimo di attrezzatura, per di più facilmente realizzabile in casa, se necessario.

1. Misurazione della resistenza serie

Sapete che questa resistenza è quella dovuta alla "imperfetta" conducibilità del rame che costituisce l'avvolgimento.
La sua accurata misurazione è fondamentale per la successiva stima del valore dell'induttanza. Purtroppo (o per fortuna, se vogliamo un'efficienza decente!), in generale:

> nelle induttanze adatte agli alimentatori switching, la resistenza serie ha un basso valore, tipicamente minore o molto minore di 1Ω, che ne rende problematica e/o imprecisa la misura diretta con un "normale" multimetro

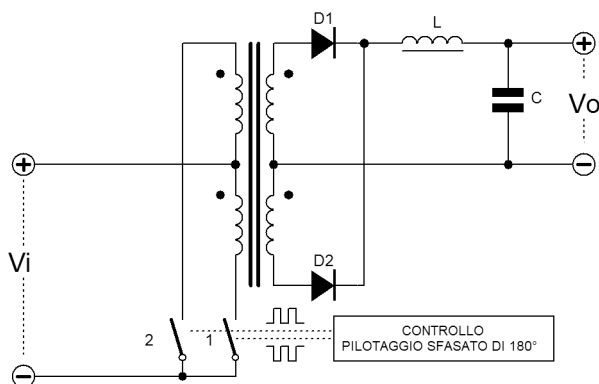



Figura 6 Tipologia a trasformatore "forward" push-pull

Solo se siete in possesso di un multimetro ad elevata risoluzione (almeno 4-5 cifre), potete misurarne il valore direttamente, avendo magari l'accortezza di sottrarre dalla misura, la resistenza dei cavi del multimetro.
Supponiamo però che questo **non** sia il vostro caso. Procuratevi quindi due resistenze di precisione, una 10Ω-5W e l'altra 100Ω-0,5W, con tolleranza dell'1% o migliore. In caso di difficile reperibilità della 10Ω-5W, consiglio di utilizzare 10 resistenze all'1%, da 1Ω ciascuna collegate in serie, oppure da 100Ω ciascuna, collegate in parallelo. Meglio, si può utilizzare una sola resistenza di precisione come la Caddock serie MP915 da 10Ω-15W, reperibile sul catalogo RS con il codice 320-4744, montata naturalmente su di una piccola aletta di raffreddamento. Realizzate poi il circuito di figura 8.

Regolate la tensione di uscita dell'alimentatore a

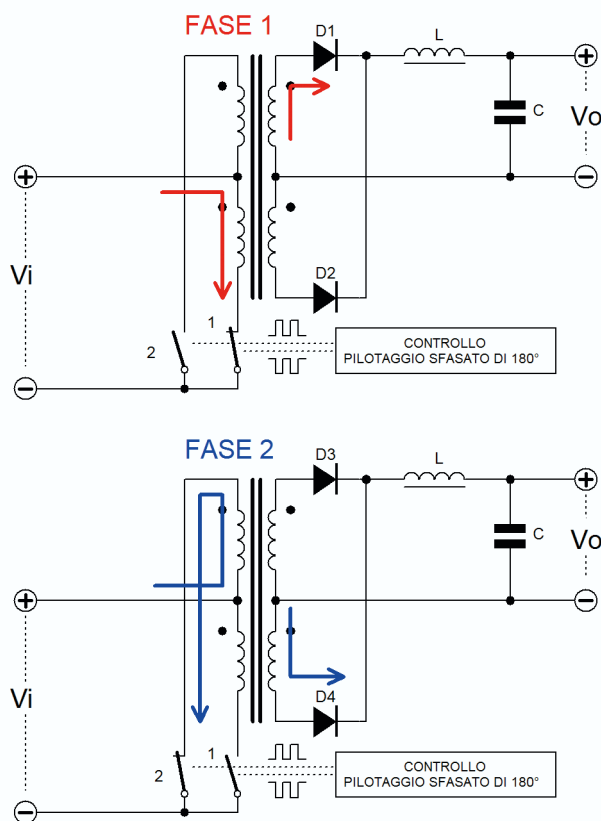



Figura 7 Ttipologia "forward" push-pull: fasi di funzionamento

circa 5V (non è importante l'esatto valore). Effettuate quindi due misure di tensione in continua con un multimetro, la prima ai capi della resistenza "campione" da 10/100Ω e la seconda ai capi dell'induttanza sotto test. Applicando la legge di Ohm, con la prima misura otteniamo la corrente che scorre nel circuito serie:

$$I = V_{MISURA1} / 100 \text{ (switch in 1)}$$
$$I = V_{MISURA1} / 10 \text{ (switch in 2)}$$

Dalla seconda misura, è facile verificare che:

$$R_L = V_{MISURA2} / I$$

Sostituendo in quest'ultima, la corrente I ricavata dalla prima espressione, otteniamo le formule che ci forniscono il valore di R^L:

$$R_L = V_{MISURA2} / V_{MISURA1} * 100 \text{ (switch in 1)}$$
$$R_L = V_{MISURA2} / V_{MISURA1} * 10 \text{ (switch in 2)}$$

Ovviamente, se si dispone di un secondo multimetro, si può evitare la prima misura collegandolo in serie al circuito e leggendo direttamente il valore della corrente.
Tenete però presente che la totalità dei multimetri presenta la migliore precisione per le misure di tensione rispetto a quelle di corrente: a buon intenditor...
Nella posizione 1, la corrente di test è pari al massimo a 50mA, e sale a 500mA nella posizione 2.
Come consiglio generale, partite sempre dalla posizione 1 e solo nel caso che la risoluzione del vostro multimetro non sia sufficiente a fornire una misura apprezzabile ai capi dell'induttanza, passate alla posizione 2. Infine, è evidente che, variando la tensione di alimentazione, è possibile variare la corrente di test. Ad esempio, se Valimentazione = 10V:

$$I_{MAX} = 100\text{mA (switch in 1)}$$
$$I_{MAX} = 1000\text{mA (switch in 2)}$$

In questo caso occorrono però resistenze in grado di dissipare una potenza maggiore; lascio a voi i relativi calcoli come semplice esercizio.

Eseguiamo ora un paio di prove su componenti reali.

Esempio 1: induttanza Panasonic serie ELC18 220µH-2,4A.
Nella posizione 2 ho ottenuto:

$$V_{MISURA1} = 5{,}058\text{V}$$
$$V_{MISURA2} = 0{,}4744\text{V}$$

da cui:

$$R_L = 93{,}8\text{m}\Omega$$

Il datasheet del costruttore specifica per questa induttanza una R_L nominale pari a 0,090Ω, con tolleranza ±20%, in ottimo accordo con la misura effettuata.

Esempio 2: induttanza C&D Technologies tipo 1422311, 22µH-11A
Nella posizione 2 ho ottenuto:

$$V_{MISURA1} = 4{,}996\text{V}$$
$$V_{MISURA2} = 0{,}0064\text{V}$$

da cui:

$$R_L = 12{,}8\text{m}\Omega$$

Il datasheet del costruttore specifica per questa induttanza una R_L massima pari a 0,011Ω; la nostra misura è comunque accettabile considerando le tolleranze del sistema (multimetro + resistenze).

Per finire, un metodo alternativo ma distruttivo per l'induttanza, consiste nello "svolgere" l'avvolgimento e misurarne la lunghezza.
Misurate poi il diametro del filo con un calibro o un micrometro, e calcolate la resistenza del filo con la formula 2, valida per il rame normalmente utilizzato negli avvolgimenti (cosiddetto rame "ricotto"), alla temperatura ambiente di 20°C.

CPLD by example è un corso introduttivo sulle CPLD suddiviso in sei puntate. Il corso è organizzato in Moduli che trattano argomenti riguardanti le Logiche Programmabili. Vengono presentati numerosi esempi pratici con circuiti applicativi e relative descrizioni in linguaggio Verilog.

Un'occasione irripetibile per comprendere il funzionamento di questi fantastici dispositivi ed utilizzarli con successo nelle vostre applicazioni!

Per non perdere nessuna puntata del corso, abbonati subito!

INDICE DELLE USCITE

Formula n.2
$R_L = \rho$ * lunghezza / sezione, cioè
$R_L = \rho$ * lunghezza / [(diametro / 2)^2 * π)]

Dove ρ è la **resistività**, che per il rame vale 1,7241 * 10^{-8} [Ω * m].

Per fare un esempio numerico, la resistenza di un avvolgimento di rame lungo 2m, di diametro pari a 0,4mm è pari a:

$R_L = 1{,}7241 * 10^{-8} * 2 / [(0{,}4 * 10^{-3} / 2)^2 * 3{,}1415] = 0{,}274\Omega$ (a 20 °C)

Tale resistenza aumenta con la temperatura secondo la formula 3.

Formula n.3
R_L (a temp. T) = R_L (a 20°C) * [1 + α * (T – 20)]

Dove α è il **coefficiente di variazione della resistività con la temperatura**, che per il rame vale 0,00393.

Per questo motivo, in tutti i casi normali, per i quali è accettabile che la temperatura dell'avvolgimento possa salire (induttanza/trasformatore dimensionati correttamente e non sovradimensionati), è bene considerare un aumento di temperatura di circa 30°C, rispetto alla massima ambiente, imposta dalle specifiche dell'alimentatore. Per esempio, supponendo di utilizzare l'induttanza Panasonic ELC18 da 220µH-2,4A in un alimentatore switching che

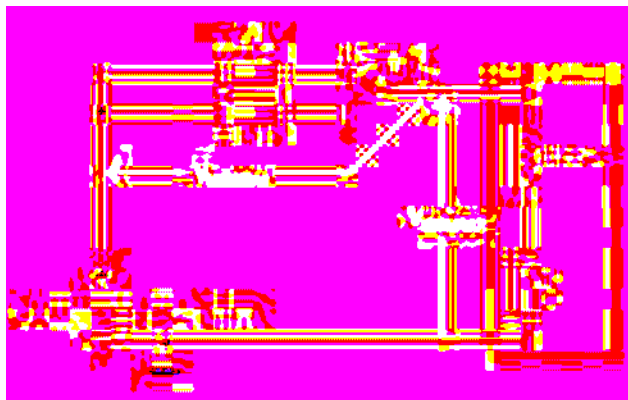



Figura 8 Test setup per la determinazione della resistenza R_L

deve funzionare correttamente fino alla temperatura ambiente di 40 °C, si può calcolare:

temp. avvolgimento = temp. ambiente max. + 30 = 70 °C

$R_{L,MAX}$ (a 70°C) = 0,0938 * (1 + α· * (70 – 20) = 0,118Ω

Questo è il valore di sicurezza da utilizzare in tutti i calcoli.

2. Misurazione dell'induttanza

Questa misurazione è leggermente più complessa. Per prima cosa occorre avere a disposizione un generatore sinusoidale: se non fosse il vostro caso, cosa aspettate a montarvene uno? Sicuramente su qualche vecchio numero di Fare Elettronica, troverete qualche oscillatore di BF a ponte di Wien!
Come secondo step, dovete conoscere il valore della resistenza (o impedenza) di uscita del vostro generatore. Preoccupati?
Ma no, se il vostro apparato è commerciale in genere è sempre riportata sul pannello, vicino al connettore di uscita. Valori tipici sono 50Ω oppure 600Ω. In molti generatori autocostruiti può essere addirittura 0Ω o quasi, in quanto si utilizza l'uscita diretta di un op-amp, che, a causa della controreazione, presenta una resistenza di uscita molto bassa. Comunque, niente paura, se necessario seguite la procedura seguente:

1. Regolate l'oscillatore a circa 1kHz, alla massima ampiezza.
2. Misurate l'ampiezza a vuoto tramite l'oscilloscopio o meglio un multimetro in AC.
3. Collegate all'uscita un trimmer multigiri da 100Ω.
4. Regolate il trimmer fino a leggere sul multimetro metà della tensione a vuoto letta al passo 2. Se non riuscite ad arrivare alla metà neanche alla minima regolazione, sostituite il trimmer con uno da 1000Ω.
5. Rimuovete il trimmer e misuratene il valore di resistenza: questa coincide con la resistenza di uscita del vostro generatore.

6. Attaccate un'etichetta con il valore di Z_{OUT} al vostro generatore!

NOTA: se il vostro generatore ha una resistenza di uscita pari a circa 0Ω, prima di proseguire provvedete ad inserire una resistenza da 50Ω (due da 100Ω-1% in parallelo) in serie all'uscita. La misurazione dell'induttanza si può effettuare con il circuito di test riportato in figura 9.

L'induttanza reale, costituita dalla serie dell'induttanza ideale L con la resistenza R_L, presenta un'**impedenza** complessiva che indicheremo con Z_L, in generale numero complesso. Per i neofiti diciamo che l'impedenza è assimilabile ad una resistenza, o meglio, la resistenza è un caso particolare dell'impedenza (quando la parte immaginaria vale zero). Anche per le impedenze vale la legge di Ohm, ed applicando la nota regola dei partitori di tensione possiamo scrivere che:

$$Vx = Z_L / (Z_L + Rs) * Vs$$

$Z_L = j * \omega * L + R_L$, indica l'impedenza complessa dell'induttanza reale; ω (lettera greca omega minuscola) rappresenta semplicemente il prodotto (2 * π * frequenza) e j è l'unità immaginaria. Nella precedente equazione, tutti i termini tranne l'induttanza L sono noti (ω, Rs) oppure misurabili (Vx, Vs, R_L).
La Vs è semplicemente la tensione a vuoto misurata ai capi del generatore. La soluzione dell'equazione rispetto a L è riportata nella formula 4, di cui ometto la noiosissima derivazione.

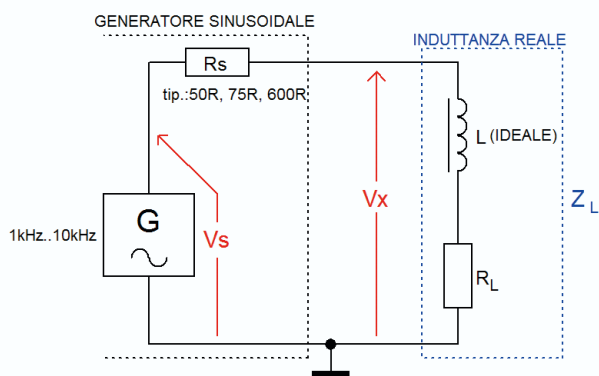



Figura 9 Test setup per la determinazione delll'induttanza L

Formula n.4
$$L = ((R_L^2 * Vx^2 - R_L^2 * Vs^2 + Rs^2 * Vx^2 + 2 * Rs * R_L * Vx^2)^{0,5}) / (2 * \pi * f * (Vs^2 - Vx^2)^{0,5})$$

La frequenza f sarebbe in teoria a piacere, tuttavia per "piccoli" valori di induttanza (<1mH) consiglio di utilizzare almeno 10kHz, mentre per valori più elevati (>= 1mH) vanno bene anche frequenze inferiori, come 1kHz. Se non conoscete a priori il valore dell'induttanza, ripetete la misura con più frequenze, in modo che con lo strumento a vostra disposizione, la misura della Vx sia sufficientemente accurata. Questo perchè, a parità di induttanza, all'aumentare della frequenza, aumenta Z_L e quindi la tensione Vx.
Applicando la formula 4 ad un caso reale, e cioè all'induttanza Panasonic ELC18 220µH-2,4A, ho ottenuto (a frequenza f = 10kHz):

$$Vs = 1,1861V$$
$$Vx = 0,2057V$$

La resistenza di uscita del mio generatore HP3336A è pari a 75Ω, mentre la R_L, calcolata nel paragrafo precedente, era di 0,094Ω.
Sostituendo tali valori nella formula 4, si ottiene:

$$L = ((0,094^2 * 0,2057^2 - 0,094^2 * 1,1861^2 + 75^2 * 0,2057^2 + 2 * 75 * 0,094 * 0,2057^2)^{0,5}) / (2 * 3,1415 * 10000 * (1,1861^2 - 0,2057^2)^{0,5})$$

cioè:

$$L = 210\mu H$$

in ottimo accordo con il valore nominale di 220µH±10%.

IL CIRCUITO INTEGRATO PWM SG3525

Introduciamo ora un "nuovo" componente che implementa un completo sistema di con-

trollo PWM V-Mode, così come il TL5001 visto in precedenza. Sto parlando del SG3525, e le virgolette sul "nuovo" si riferiscono al fatto che in realtà, è solo un aggiornamento del mitico 3524, uno dei primi (se non il primo) circuiti integrati PWM, introdotto nel lontano 1976! A partire da questo capostipite, negli anni si sono succedute diverse versioni allo scopo di supplire ai difetti e di migliorare le caratteristiche della prima versione. In figura 10 è riportato lo schema a blocchi interno.

Balzano subito all'occhio le aggiunte rispetto al più semplice TL5001:

1. Ci sono due uscite complementari, A e B.
2. Le uscite sono del tipo "totem-pole", e riescono a pilotare attivamente un carico sia verso massa che verso Vcc. Ricordo che il TL5001 ha un'uscita di tipo open-collector, e si affida ad una resistenza esterna per "sollevare" il carico verso Vcc.
   Le uscite totem-pole permettono all'SG3525 di fare a meno del classico circuito buffer push-pull costituito da un NPN e da un PNP, che abbiamo utilizzato finora.
3. Sia la resistenza RT che il condensatore CT di temporizzazione dell'oscillatore sono esterni, il che rende possibile l'utilizzo di componenti a bassa tolleranza per ottenere una frequenza ben precisa. Ricordo che il TL5001ha il condensatore integrato, di tolleranza relativamente ampia.
4. Esiste un piedino di ingresso di sincronizzazione, utile ad esempio per ridurre i disturbi (accenneremo più avanti come), ed anche l'uscita dell'oscillatore è disponibile, per sincronizzare altri dispositivi esterni.
5. Sia il riferimento interno, che entrambi i piedini dell'amplificatore di errore, sono portati esternamente per la massima versatilità.
6. È implementata internamente la funzione di shutdown.

Lo scotto da pagare è ovviamente un maggior ingombro, infatti l'SG3525 è contenuto in un DIP16/SO16, a differenza del TL5001 cui è sufficiente un DIP8/SO8.

CONCLUSIONI

Concludiamo qui, per ora, la breve presentazione dell'SG3525.

Nella prossima puntata utilizzeremo questo componente come cuore di un alimentatore switching forward di tipo push-pull, e realizzeremo il nostro primo trasformatore per switching!
Chi volesse può sin d'ora procurarsi un nucleo completo E20, in ferrite N27 o similare.
Per chiarimenti contattatemi pure via e-mail...vi aspetto alla prossima puntata.

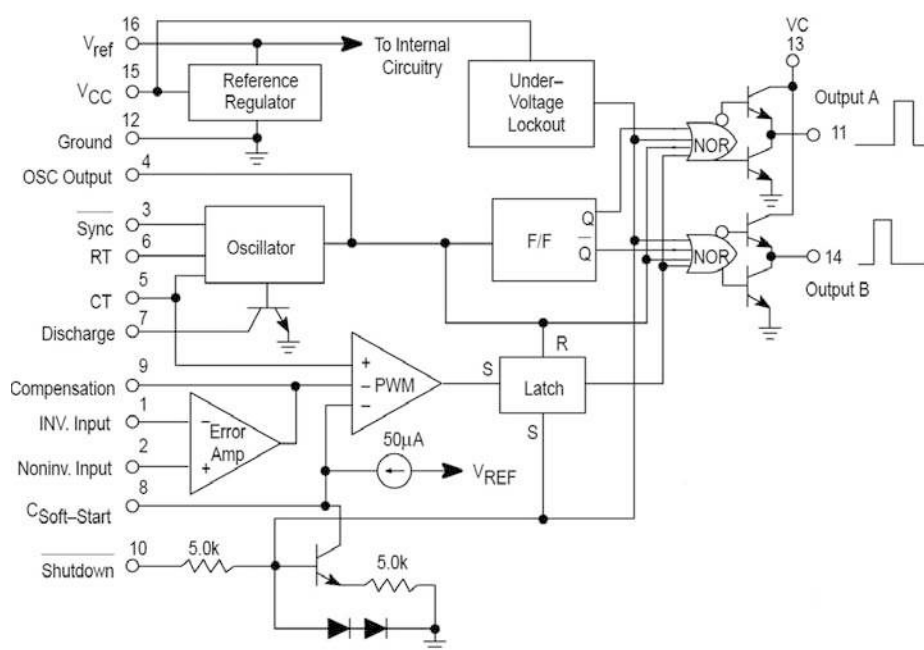



Figura 10 *Schema a blocchi interno del SG3525*

Vitamina C:

In questa puntata verrà introdotta l'aritmetica fixed point, una tecnica che permette di manipolare numeri frazionari utilizzando soltanto tipi interi (senza bisogno di ricorrere alla pesante rappresentazione floating point). Grazie a questa tecnica sarà possibile riscrivere in maniera estremamente più efficiente molti algoritmi tradizionalmente basati su floating point, e renderne possibile la loro esecuzione perfino su piccoli microcontrollori privi di FPU.

In molte applicazioni può presentarsi la necessità di manipolare numeri non interi, cioè numeri "reali" o che comunque comprendono una parte frazionaria. Spesso quando si sviluppa un programma in C l'approccio più semplice è quello di usare variabili di tipo floating point per trattare queste grandezze. Questa soluzione sebbene molto comoda (dal punto di vista di un programmatore ad alto livello), ha una serie di implicazioni non indifferenti in pratica, che devono essere invece considerate se si vuole ottenere un'implementazione efficiente. Innanzi tutto i calcoli che coinvolgono numeri floating point richiedono un tempo di esecuzione sempre sensibilmente più grande delle equivalenti operazioni svolte in aritmetica intera. Secondariamente la memorizzazione dei numeri floating point nella forma più semplice supportata dall'ANSI C (float) richiede 32 bit, una quantità di memoria che in molti casi è superiore a quella effettivamente richiesta dai calcoli che si stanno eseguendo. L'aumento dei tempi di elaborazioni, e della quantità di risorse di memoria richieste possono essere notevoli, perfino su macchine dotate di grandi risorse e capacità di calcolo, come i normali PC. Ovviamente la situazione risulta ancora più grave quando il programma che si sta scrivendo dovrà funzionare su dei piccoli sistemi embedded, tipicamente basati su microcontrollore. In questi casi infatti raramente si può contare sull'aiuto di un'unità a virgola mobile (FPU), e quindi le operazioni sui tipi float saranno svolte da apposite librerie software, con un enorme incremento dei tempi di esecuzione. Un ulteriore svantaggio a cui spesso non si presta molta attenzione è rappresentato dal fatto che l'uso di aritmetica floating point comporta un maggiore consumo di potenza! Questo si traduce in un maggiore assorbimento di corrente, un conseguente aumento della dissipazione termica, ed in generale una riduzione dell'autonomia di eventuali batterie. Come se non bastasse infine è abbastanza difficile convertire un numero rappresentato in floating point in ASCII, a meno di non usare librerie dedicate ed in genere abbastanza "voluminose".
Come vedremo di seguito nella maggior parte dei casi è possibile fare a meno dell'uso dell'aritmetica floating point, approssimandola con quella intera. I vantaggi offerti da questa tecnica possono essere enormi, basti pensare ad esempio alla possibilità di utilizzare funzioni trigonometriche su processori capaci di manipolare solo numeri interi. La tecnica di cui stiamo parlando, e che verrà descritta nei successivi paragrafi, si chiama aritmetica a "virgola fissa" (*fixed poin*t in inglese). I dettagli su come

di Antonio Di Stefano
a.distefano@farelettronica.com

Routine matematiche aritmetica fixed point

implementare ed utilizzare in C questa tecnica verranno dati nella prossima puntata.

FIXED POINT IN BASE 10

Per comprendere il funzionamento dell'aritmetica fixed point sarebbe sufficiente rispolverare un po' della teoria riguardante la rappresentazione dei numeri nelle diverse basi. Tuttavia possiamo raggiungere lo stesso scopo in maniera più intuitiva ricorrendo a degli esempi ed iniziando dai più familiari numeri decimali. Immaginiamo di dovere leggere il valore fornito da un sensore di temperatura e di doverlo visualizzare su un display in gradi Celsius. Supponiamo che il range di temperatura in cui opera il sensore sia compreso tra 0°C e 60°C, e fornisca in questo intervallo un valore ad 8 bit (quindi compreso tra 0 e 255). Per ottenere una indicazione in gradi Celsius sarebbe necessario moltiplicare il valore fornito dal sensore per 0.235 (cioè 60/255). La moltiplicazione per un numero frazionario apparentemente richiederebbe l'uso della rappresentazione in virgola mobile. In realtà possiamo notare che la cifra visualizzata sarà sempre compresa tra "00.00" e "60.00". Perché allora non utilizzare un numero intero compreso tra 0 e 6000 per rappresentarla? (Visualizzando il punto decimale tre le prime due cifre e le seconde, ed aggiungendo eventualmente degli zeri in testa). Se il dato di partenza è un intero, ed anche il risultato è un intero, possiamo pensare di utilizzare soltanto l'aritmetica intera per eseguire i calcoli! Il risultato può essere ottenuto ad esempio utilizzando la seguente espressione:

```
T = (Dato * 235)/10
```

Ovviamente si tratterà di un valore approssimato, ma comunque accettabile per l'applicazione.
Cerchiamo di schematizzare meglio quanto visto sopra ed estenderlo al caso generale. L'aritmetica fixed point decimale è utile quando gli operandi o i risultati sono originati o devono essere convertiti in formati visualizzabili su un qualche tipo d'interfaccia utente (ASCII, BCD, 7 segmenti…). Come mostrato nell'esempio il procedimento consiste nel considerare l'espressione decimale del numero, e moltiplicarlo per potenze di 10 in modo da renderlo intero. A questo punto è possibile eseguire le operazioni utilizzando la normale aritmetica intera, ed eventualmente riportare il risultato al formato originario piazzando il punto decimale dopo la cifra stabilita. In realtà le cose non sono così immediate come potrebbe sembrare da questa descrizione, infatti nel corso delle operazioni è sempre necessario tenere traccia "manualmente" della posizione del punto decimale, e fare in modo che la sua posizione negli operandi sia correttamente allineata. Di solito si usa la notazione S.X.Y, per indicare che il numero è dotato di segno (S), di X cifre a sinistra del punto decimale (intere), e di Y cifre a destra (frazionarie). Se ad esempio consideriamo un numero composto da 5 cifre intere (in base 10), il suo formato sarà indicato come S.5.0. Mentre ad esempio un numero composto da 3 cifre intere e 2 decimali sarà indicato come S.3.2. Normalmente il numero di cifre complessivo utilizzabile è fissato, ad esempio un numero intero a 16 bit (uno *short int*) avrà al massimo 5 cifre (in base 10). Queste potranno essere utilizzate per rappresentare sia la parte intera che quella frazionaria, fissando implicitamente la posizione del punto decimale. Alcuni esempi di "conversione" sono mostrati di seguito, considerando una rappresentazione S.3.2:

```
5.81 => 581
-142.25 => -14225
```

```
9.3 => 930
```

Per ottenente dei numeri interi non abbiamo fatto altro che moltiplicare per 10 elevato al numero di cifre frazionarie (in questo caso 10^2=100). Questi numeri, essendo nello stesso formato possono essere sommati o sottratti con i normali operatori interi, supportati direttamente a qualsiasi processore.
Che succede se dobbiamo sommare o sottrarre due numeri espressi in formato diverso? Occorre prima allineare la posizione dei loro punti decimali, ossia ricondurli allo stesso formato. Ad esempio, se consideriamo il numero 91.21 in formato S.3.2, e 54.1 in formato S.4.1 e vogliamo sommarli abbiamo:

```
91.21 => 9121 (S.3.2)
24.1  =>  241 (S.4.1)
```

Se guardiamo i numeri interi ottenuti notiamo che per allinearli correttamente è necessario moltiplicare per 10 quello in formato S.4.1. Questo risulta chiaro se si pensa al fatto che il punto implicito del primo si trova prima delle cifre "21", mentre nel secondo si trova prima dell'"1". Quindi i numeri da sommare sono 9121 e 2410 (ora entrambi nel formato S.3.2), che forniscono un risultato pari a 11531, che va interpretato anch'esso come numero in formato S.3.2, cioè 115.31.
E' possibile estendere i ragionamenti fatti anche alle altre operazioni aritmetiche. Il procedimento sarà comunque spiegato in maniera più completa nel prossimo paragrafo, con riferimento all'aritmetica fixed point binaria.

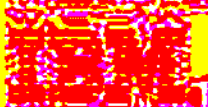
FIXED POINT IN BASE 2

Il fatto di dovere moltiplicare o dividere per potenze di 10 per allineare i numeri, rende l'aritmetica fixed point decimale poco efficiente, infatti non tutti i processori supportano direttamente le operazioni di moltiplicazione e divisione. Per superare questo limite occorre ripensare in base 2 gli stessi ragionamenti fatti prima: le cifre da considerare saranno i bit che compongono il numero, e le potenze per cui moltiplicare e dividere saranno quelle di 2. Quest'ultimo particolare risulta molto importante, infatti moltiplicare un numero binario per potenze di due significa semplicemente scorrere i suoi bit a destra o a sinistra! Lo svantaggio in questo caso è che il numero intero che si ricava, visivamente non somiglierà affatto all'originale, quindi risulterà più difficile da visualizzare su un display, ma sarà comunque adatto ad eseguire i calcoli intermedi di un algoritmo e risulterà anche più preciso di quello decimale. Proprio per la maggiore efficienza, l'aritmetica fixed point binaria è in effetti la più utilizzata in pratica. Supponiamo di utilizzare ancora una volta numeri a 16 bit. Questa volta la corrispondenza tra cifre e bit è diretta, quindi se ci riferiamo al formato S.7.8 stiamo considerando un numero binario dotato di un bit di segno, 7 bit per la parte intera ed 8 per quella frazionaria. Ad esempio, i numeri 49.3 e 19.73 possono essere espressi in formato fixed point nel seguente modo:

```
19.73 =>  19.73*2⁸ =   5051 (S.7.8)
-49.3 => -49.3 *2⁸ = -12621 (S.7.8)
```

Per ottenere questi numeri interi è stato sufficiente moltiplicare per 2 elevato al numero dei bit frazionari, cioè 8 in questo caso. I due numeri ottenuti possono essere sommati o sottratti, ottenendo ancora un numero intero, che deve essere interpretato nel formato S.7.8 (cioè va diviso per 2^8=258 per riottenere l'equivalente decimale):

```
  5051+ (S.7.8)
-12621= (S.7.8)
 -7570  (S.7.8)
```

Dividendo -7570 per 256 si ottiene -29.57, che è il risultato corretto della somma dei due numeri originali. In questi calcoli è stato necessario troncare o arrotondare i risultati parziali. Questo è dovuto al fatto che la parte frazionaria decimale necessita in genere di un numero di cifre binarie frazionarie maggiore per essere rappresentata. Se abbiamo a disposizione n cifre binarie per la parte frazionaria l'errore di rappresentazione sarà dell'ordine dell'LSB, cioè di $1/2^n$ (ad esempio nel caso di 8 cifre sarà di circa 1/256 = 0.0039). Questo errore può essere ridotto di metà se prima di troncare il numero si somma il valore di ½LSB (cioè

LinuxWorld
Per info e registrazioni
www.linuxworldexpo.it
Tel. 02 48100906
info@linuxworldexpo.it

$1/2^{n+1}$). Non tratteremo qui le tecniche di arrotondamento per questioni di spazio. Come nel caso della rappresentazione fixed point decimale, anche in binario è necessario fare in modo che i due numeri siano allineati prima di eseguire le operazioni di addizione o sottrazione. Se ad esempio abbiano il numero 5.41 in formato S.7.8, ed il numero 1.243 in formato S.5.10, allora sarà necessario moltiplicare il primo per 2 elevato a 2 (10-8=2), cioè 4, oppure dividere il secondo per lo stesso fattore. Il risultato si otterrà in formato S.5.10 nel primo caso, o S.7.8 nel secondo. Di seguito sono mostrati i due casi:

```
5.41 * 256 = 1385    (S.7.8)
1.243 * 1024 = 1273  (S.5.10)
1385*4 + 1273 = 6813 (S.5.10)
6813 / 1024 = 6.653  (dec)
```

oppure

```
5.41 * 256 = 1385    (S.7.8)
1.243 * 1024 = 1273  (S.5.10)
1385 + 1273/4 = 1703 (S.7.8)
1703 / 256 = 6.652   (dec)
```

La scelta di uno dei due formati dipende da quale parte del numero sia più significativa dal punto di vista dell'applicazione (il primo ad esempio conserva una maggiore precisione sulle cifre frazionarie).

Moltiplicazione

Consideriamo adesso l'operazione di moltiplicazione tra numeri fixed point. Questa operazione è forse la più interessante, perché è la più utile nella maggior parte della applicazioni. Purtroppo però è anche più delicata della altre da gestire. Il problema della moltiplicazione è che il numero di bit del risultato aumenta rispetto a quello degli operandi, ed anche il formato risulta modificato. In particolare il prodotto tra due numeri fixed point di formato S.X.Y ed S.Z.W avrà il formato SS.(X+Z).(Y+W). Saranno cioè presenti due bit di segno, e le parti intere e frazionarie saranno ampie come la somma delle ampiezze delle rispettive parti degli operandi. Questo ha due conseguenze principali: 1) i risultati intermedi devono essere contenuti in variabili dotate di un numero di bit maggiore di quelle di partenza (tipicamente larghe il doppio); 2) sono necessarie delle operazioni di troncamento/arrotondamento e divisione per riportare il risultato al formato originario. Facciamo un esempio per capire meglio, considerando la coppia di numeri 5.41 e 1.243 visti sopra, espressi in formato S.7.8:

```
  1385x (S.7.8)
   318= (S.7.8)
440430  (SS.14.16)
```

Per riportare il risultato al formato di partenza (S.7.8) è necessario troncare la parte frazionaria ai primi 8 bit (scorrendo a destra di 8 bit, cioè dividendo per 256), e assicurarsi che la parte intera non necessiti di più di 7 bit per essere rappresentata (i bit in più verranno troncati dal momento che il risultato sarà collocato in una variabile a 16 bit):

```
440430/256 = 1720 (S.7.8)
1720/256 = 6.718  (dec)
```

Se i due operandi hanno formati diversi non ci sono particolari problemi, le dimensioni delle due parti del risultato saranno sempre date dalla somma di quelle degli operandi, per cui sarà sufficiente scegliere un opportuno fattore di divisione per riportare il risultato al formato originario. Ad esempio:

```
 (S.7.8)x
(S.5.10)=
(SS.12.18)/2^10=(S.7.8)
```

Divisione

Utilizzando l'aritmetica fixed point è possibile eseguire in due modi diversi le divisioni. Uno di questi è paragonabile al metodo tradizionale, l'altro invece risulta molto più vantaggioso, quando è applicabile.
Se consideriamo la divisione normale, notiamo che si verifica subito un effetto indesiderato, essa infatti ha l'effetto di cancellare i fattori costanti che ci hanno permesso di rendere interi i numeri su cui vogliamo operare! Cioè si ha che:

```
X*2^n/Y*2^n = X/Y
```

Se, come supporto all'inizio, possiamo effettuare soltanto divisioni intere, otterremo ogni volta la perdita di tutti i bit frazionari! Ad esempio:

```
1385/ (S.7.8)
 318= (S.7.8)
   4  (S.7.0)
```

Per evitare questo effetto occorre moltiplicare il dividendo per una potenza di due per compensare la perdita dei bit, questa sarà pari al numero di bit frazionari che si vogliono ottenere:

```
1385*256 = 354560 (S.7.16)

354560/ (S.7.16)
    318= (S.7.8)
  1115  (S.7.8)
```

Il problema in questo caso è che per memorizzare il dividendo "allargato" occorrono utilizzare variabili più grandi di quelle degli operandi (tipicamente il doppio), un po' come succede per la moltiplicazione. Un altro problema caratteristico di questo approccio (che si ha anche lavorando con semplici numeri interi), è che la divisione, anche quando è supportata dall'hardware risulta piuttosto lenta rispetto alle altre operazioni. Questo problema può essere superato utilizzando un metodo diverso per eseguirla. Infatti se dobbiamo dividere un numero R per una costante S, possiamo ottenere lo stesso risultato moltiplicando R per 1/S. Ad esempio, se vogliamo sapere a quanti giorni corrispondono 105 ore, anziché dividere 105 per 24, possiamo moltiplicarlo per 1/24=0.041666. Se utilizziamo una rappresentazione S.7.8 per entrambi i numeri abbiamo:

```
105*2^8= 26880   (S.7.8)
0.041666*2^8= 11 (S.7.8)
26880*11 = 295680 (SS.14.16)
295680/2^8 = 1155 (S.7.8)
1155/2^8 = 4.51   (dec)
```

Il risultato corretto sarebbe 4.375, quindi il valore ottenuto soffre un po' del ristretto numero di bit utilizzati nei diversi passi per rappresentare la parte frazionaria. Visti i numeri in gioco avremmo potuto utilizzare dei formati tagliati su misura per i rispettivi operandi, ad esempio S.15.0 per il primo, dal momento che è già un numero intero, ed S.0.15 per il secondo, che invece è puramente frazionario. Il risultato è il seguente:

```
105                  (S.15.0)
0.041666*2^15 = 1365 (S.0.15)
105*1363 = 143325  (SS.15.15)
143325/2^7 = 1120   (S.7.8)
1120/256 = 4.375   (dec)
```

Questo esempio mette in luce una cosa importante: la scelta del formato da utilizzare in un determinato algoritmo deve tenere conto delle caratteristiche dei numeri che si devono trattare, della possibilità che si verifichino degli overflow, della disponibilità di memoria e del supporto da parte dell'hardware per l'esecuzione delle operazioni nei vari formati. In questo caso abbiamo utilizzato sempre numeri a 16 bit, ed a 32 bit per i risultati intermedi, entrambi con segno. Se si hanno esigenze di precisione meno stringenti è possibile utilizzare anche tipi ad 8 ed a 16 bit (ad esempio nei formati S.6.2, S.3.4, S.0.7, S.7.0, etc.).
Va ricordato infine che è possibile applicare le tecniche viste anche a numeri privi di segno, l'unica differenza consiste nel fatto di si ha a disposizione un bit in più, che può essere sfruttato per estendere la larghezza dell'intervallo rappresentato (parte intera) o la sua precisione (parte frazionaria).

Q-FORMAT

Se è necessario eseguire molte operazioni di moltiplicazione di seguito sugli stessi numeri (operazioni di accumulazione), può capitare che la parte intera cresta via via fino a non essere più rappresentabile col numero di bit di cui si dispone. Per evitare questo problema si può procedere in due modi: o si dividono i risultati intermedi per 2 (eseguendo degli scorrimenti a destra), o si utilizzano numeri privi della parte intera, quindi compresi nell'intervallo [-1,1]. La scelta di una delle due soluzioni dipende ovviamente dal particolare

algoritmo che si sta implementando. In questo paragrafo ci soffermeremo sul secondo metodo, spesso indicato col nome di "Q-format", dal momento che esso è di gran lunga il più utilizzato. Potere rappresentare delle grandezze nell'intervallo [-1,1] risulta fondamentale in moltissime applicazioni, infatti permette ad esempio di rappresentare segnali normalizzati di diverso tipo (audio, video…), ma anche funzioni trigonometriche come seno e coseno.
Per questo motivo il Q-format è supportato direttamente in hardware da molti DSP. Rispetto a quanto detto prima il Q-format non introduce particolari novità: esso consiste in una rappresentazione del tipo S.0.N, cioè in cui tutti i bit, tranne quello di segno, sono utilizzati per rappresentare la parte frazionaria. Questo fa sì che in realtà l'intervallo rappresentabile sia compreso tra -1 e poco meno di +1. Uno dei formati più utilizzati è il già citato S.0.15 (detto anche Q-15).
La somma e la sottrazione tra numeri in Q-format è identica al caso intero, infatti si possono utilizzare i normali operatori interi, a differenza di quanto visto prima tutti i numeri risultano sempre allineati, ma esiste la possibilità che si verifichino degli overflow se il risultato supera i limiti rappresentabili. La moltiplicazione invece, come già accennato, normalmente non genera overflow! Questo è dovuto al fatto che quando si moltiplicano due numeri appartenenti all'intervallo [-1,1], il risultato appartiene ancora a questo intervallo! Esiste solo un'eccezione che si verifica quando si moltiplica -1 x -1. Il risultato dovrebbe essere 1, ma come abbiamo visto questo numero non è rappresentabile. Il più grande numero positivo (nel caso di Q-15) infatti è $(2^{15}-1)/2^{15} = 0,99996$.
Per eseguire la moltiplicazione tra due numeri Q-15 si ricorre al seguente procedimento: si moltiplicano i due numeri S.0.15 (16 bit) ottenendo un numero SS.0.30 (32 bit), come già visto. Come si vede il risultato è anch'esso costituito soltanto da bit frazionari (a parte i due bit di segno), questo significa che possiamo trascurare i 15 bit meno significativi, ottenendo lo stesso numero frazionario, ma un po' meno preciso (15 bit invece che 30). Per fare questo, e riportare il numero al formato originario a 16 bit è quindi sufficiente scorrere a destra di 15 bit il risultato, e considerare solo i 16 bit meno significativi (oppure scorrere di un bit a sinistra, per cancellare il bit di segno in più, e considerare i 16 più significativi). Esempio:

```
 0.69376x (dec)
-0.50584= (dec)
-0.35093  (dec)

 0.69376x2^15 =  22733 (S.0.15)
-0.50584x2^15 = -16575 (S.0.15)

      22733x
     -16575=
-376799475    (SS.0.30)
        /2^15=
     -11499= (S.0.15)
  -0.35092  (dec)
```

La divisione in Q-format crea qualche problema, infatti è abbastanza comune che due numeri contenuti nell'intervallo [-1,1] divisi tra loro risultino in un numero che sta al di fuori di questo intervallo. Non solo, anche i reciproci stanno necessariamente fuori dall'intervallo! L'unica soluzione possibile, se non si riesce ad aggirare il problema modificando l'algoritmo, è quella passare ad una rappresentazione S.X.Y, e quindi operare come descritto prima, e ricondursi al formato Q-n alla fine, se possibile. In realtà comunque, questo è un problema mal posto, perché uno dei vantaggi di lavorare nell'intervallo [-1,1] è proprio quello di potere fare a meno della divisione! E del resto gli algoritmi che si prestano ad utilizzare questo formato raramente contengono delle divisioni esplicite.
Tutti i procedimenti visti fino a qui si possono adottare anche nel caso di rappresentazioni caratterizzate da un numero diverso di bit, ad esempio Q-3, Q-7, Q-31…

CONCLUSIONE

In questa puntata abbiamo visto come funziona l'aritmetica fixed point nei vari casi, nella prossima invece ci concentreremo sui dettagli della sua implementazione in C. In particolare verrà mostrato il codice relativo ad alcuni interessanti esempi che la utilizzano.

Quando hardware e software si incontrano...
INTERFACCIA USB per PC
K8055
montata - VM110
CATALOGO generale

Scuolabus

L'Istituto Professionale Statale per l'Industria e l'Artigianato "G. Ferraris" di Fasano (Brindisi) è nato nel 1960 come costola dell'omonimo istituto di Brindisi, da cui ancora oggi dipende pur essendo separato da una distanza di 60 kilometri.

La distanza dalla centrale ha portato la sede di Fasano a sviluppare notevoli capacità adattative, che si sono dimostrate vincenti ogni volta che negli esami finali sono stati confrontati i risultati raggiunti da classi parallele delle due sedi.
La scuola promuove la continuazione di una tradizione artigiana secolare, riconosciuta da tutti i centri vicini. In particolare tra gli scopi formativi c'è quello di conservare e affinare la diffusa passione per la meccanica che si manifesta nell'annuale competizione automobilistica Fasano-Selva.
Nonostante il ridotto numero di allievi, l'IPSIA di Fasano può vantare di essere il maggior "fornitore della real casa", cioè di essere la scuola che negli ultimi anni ha formato il maggior numero di nuovi tecnici assunti in ambito regionale dall'ENEL.
Questi sono i dati della sede di Fasano: due specializzazioni, elettrotecnica e meccanica; dotazione strumentale ridotta ed essenziale, come si conviene ad una sede staccata; progetti qualificati ed ambiziosi.
Tra i progetti di maggiore rilievo elenchiamo:

- La realizzazione di un piccolo sistema eolico-fotovoltaico con "girasole" di orientamento automatico.
- La produzione di un video di autopresentazione della scuola ("Una visita elettrizzante e dinamica").
- La realizzazione di un sistema multimediale costituito da un proiettore video, da un amplificatore audio da 120W autocostruito e da uno schermo motorizzato, utilizzabile per assemblee, proiezioni di film e lezioni collettive su computer.
- La presentazione di una mostra didattica ("Fiat lux: dalla lucerna alla lampada ad induzione. Storia, arte e tecnica dell'illuminazione") durante la Settimana della cultura scientifica.

Le visite guidate sono occasione di contatto con realtà tecniche molto evolute: le ultime uscite hanno avuto per oggetto la visita al Centro per lo studio delle onde gravitazionali a Pisa e la visita alle principali aziende motoristiche dell'Emilia Romagna.

Gli allievi, in occasione degli esami finali, vengono guidati nella stesura di approfondimenti tecnici (tesine) completi di modellini dimostrativi e

Figura 1 *IPSIA "G. Ferraris" di Fasano*

IPSIA "G. Ferraris" di Fasano (BR)

riguardanti soluzioni originali a problemi applicativi delle conoscenze.
Con il passare degli anni la scuola si è così dotata di una ricca "biblioteca" di soluzioni tecniche, che viene messa a disposizione su CD-ROM per tutti i nuovi docenti che ne fanno richiesta.
La Terza Area, affidata a esperti esterni per un totale di 300 ore annue, ha per oggetto le Tecniche di controllo ed automazione industriale.
Lo stage applicativo facente parte di detta Area riguarda "l'Autronica" (elettronica applicata all'auto) vista come simulazione delle applicazioni industriali di sensori ed attuatori. Il sito Internet dell'istituto (www.ipsiaferraris.it) contiene oltre alla classica presentazione delle strutture e dell'offerta formativa, anche una interessante sezione dedicata all'autoapprendimento con vere e proprie lezioni in rete.

Allegato a questa presentazione, sarà descritto il **Progetto EOLO** sulla realizzazione di un **generatore eolico**, che ha impegnato gli allievi nel corrente anno scoalastico.

INTERVISTA AL DIRIGENTE SCOLASTICO PROFFESSOR DOMENICO CAMARDA

Quante sedi ha il suo Istituto e quali sono gli indirizzi di studio offerti?
L'Istituto "G. Ferraris" di Brindisi dispone di due sedi: una sede centrale nel capoluogo ed una sede coordinata in provincia, a Fasano. L'Istituto è articolato per la sede di Brindisi in tre indirizzi (elettronica, elettrotecnica e meccanica) e per sede coordinata di Fasano in due (elettrotecnica e meccanica).

Figura 2 Il dirigente prof. Domenico Camarda

Ad un anno dalla sua nomina alla presidenza, quale giudizio esprime su questo Istituto?
L'istituto è ricco di potenzialità che sono emerse nel corso dell'anno precedente; ora che sono stato nominato a tempo indeterminato sarà mia cura valorizzarle al meglio.
Seguirò con particolare attenzione due progetti: il primo, nella sede di Brindisi, riguarda il "Modello di una azienda certificata ISO 9000 per lo studio, la progettazione e la produzione di attrezzature sportive". Il secondo, nella sede di Fasano, riguarda la sperimentazione di una forma di scuola aperta rivolta alle imprese artigiane locali.

Quali sono le prospettive di lavoro che l'IPSIA offre ai suoi allievi?
Il fine dell'istruzione professionale è immettere nel mondo del lavoro un "prodotto" competitivo, vista la crisi nel campo dell'occupazione. Una preparazione finale solida consente sia l'iscrizione all'università o alle accademie militari che l'immissione nel mondo del lavoro come tecnico specializzato o come libero imprenditore.

Come la scuola attualizza l'offerta formativa?
Attraverso un legame più stretto con il mondo del lavoro. Infatti il discorso dell'alternanza scuola-lavoro deve vedere meglio comunicare il mondo delle imprese con la scuola, in modo che le vicendevoli sinergie assicurino la preparazione necessaria. Con gli stage della cosiddetta Terza Area gli allievi entrano in contatto direttamente con entità produttive realmente operanti e si aggiornano sullo "stato dell'arte".

Prima parte
Generalità

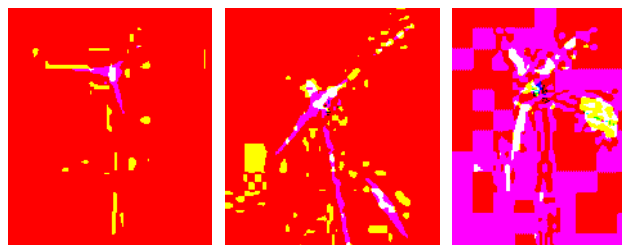
Seconda parte
L'alternatore

Terza parte
Costruzione delle pale

Costruire un generatore eolico partendo da zero

In questo articolo gli allievi del 5° anno dell' IPSIA "G. Ferraris" di Fasano, sotto la guida dei proff. Antonio Cecere e Beatrice Paparella, descrivono la loro partecipazione al "Progetto EOLO", in cui hanno sperimentato la costruzione di un generatore eolico a basso costo con mezzi alla portata di tutti.

PERCHÉ UN GENERATORE EOLICO?

Sicuramente sarà capitato a tanti proprietari di seconde case di ricevere delle consistenti bollette di telefono e di energia elettrica, pur sapendo di non aver consumato nulla nei mesi vuoti. Questo succede perché, non essendo prevista nessuna fascia sociale, per queste case i canoni sono doppi o tripli.
Molti hanno già rinunciato alla linea telefonica, optando per il più versatile telefonino, che ora, oltre a sostituire in tutto e per tutto il classico telefono, può fare molto di più.
Perché non distaccarsi anche dalla rete elettrica, creando un sistema di generazione autonomo, ad esempio ad energia eolica?
Molti ci hanno pensato, ma pochi l'hanno fatto ed il primo motivo è sicuramente l'alto costo iniziale. Il costo di un generatore eolico di piccola taglia è dovuto soprattutto al fatto che nel mondo esistono centinaia di piccoli produttori di tipo artigianale che non hanno né alti volumi di produzione né processi automatizzati.
Sicuramente, abbattendo il costo iniziale, gli interessati sarebbero molti di più. Tra gli interessati ci sono sicuramente i possessori di case isolate, tanto lontane dai centri urbani da non sperare di potersi un giorno collegare alla rete elettrica pubblica. Altrettanto interessati sono gli abitanti delle piccole isole in cui non c'è rete elettrica, oppure ce n'è una con generatore ibrido fotovoltaico–diesel che non sempre risulta affidabile.
Noi vogliamo dimostrare che è possibile autocostruire in modo facile ed economico un generatore eolico. Non è tutto frutto del nostro ingegno quello che leggerete: ci siamo serviti dell'esperienza e della collaborazione dei nostri docenti, ma soprattutto abbiamo saccheggiato

Esempi di generatori eolici autocostruiti

Gruppo di lavoro del Progetto Eolo

di Antonio Cecere
a.cecere@farelettronica.it

Generalità

la rete per eccellenza… INTERNET. Non diceva forse Newton che, per vedere lontano, era salito sulle spalle dei giganti?

QUANTA ENERGIA SI PUÒ RICAVARE DAL VENTO?

Sottrarre energia al vento è stato sempre un pallino dell'uomo: in Mesopotamia lo facevano già 2000 anni prima di Cristo per l'irrigazione; i naviganti lo hanno sempre fatto con le imbarcazioni a vela e lo facevano gli americani nella prima metà del '900 in molte fattorie (lo vediamo nei vecchi film), quando non c'era ancora l'elettrificazione rurale.
Il vento è sinonimo di imprevedibilità e di variabilità: per sapere quanto ne abbiamo a disposizione non ci resta che affidarci alle statistiche.
L'intensità, la direzione e la durata del vento infatti variano moltissimo da un posto all'altro in funzione della particolare conformazione del territorio (alberi, case, colline, montagne, coste, ecc.). Prima di installare un generatore eolico è bene conoscere la "bontà" del sito con una indagine anemologica, cioè con uno studio statistico condotto con un misuratore di vento (anemometro). Dati molto accurati sono di solito in possesso delle stazioni metereologiche e degli aeroporti. Può essere utile consultare i siti *www.ucea.it/indagro/index.htm* e *www.inea.it/otris/file/clima.htm.*
Il CESI ha presentato di recente, in collaborazione con l'Università di Genova, un Atlante Eolico dell'Italia con molte mappe particolareggiate *(www.ricercadisistema.it/html/ita.3.37/index.htm).*
La figura 1 riporta la mappa generale del vento in Italia, estratta da quest'ultimo sito.
Osservando i dati statistici citati si desume che la velocità media del vento è più alta d'inverno, quando invece i pannelli fotovoltaici sviluppano poca energia a causa della minore insolazione. Quindi l'energia eolica è un ottimo complemento dell'energia fotovoltaica per avere una disponibilità di energia pressoché costante per tutto l'anno. Generatori eolici e pannelli fotovoltaici possono concorrere senza problemi a caricare lo stesso sistema di accumulatori.

Perché un generatore eolico possa generare corrente, deve essere investito da un vento di velocità superiore a 2,5 – 3 m/s, detta in termini tecnici "velocità di cut-in". All'opposto, una velocità superiore a 15 m/s potrebbe danneggiare il generatore e perciò a questa velocità (detta di "cut-off") devono intervenire dei meccanismi di protezione.
La potenza posseduta dal vento dipende dalla sua velocità e dall'area del generatore investita:

$$Pv = 0,5 \bullet \rho \bullet A \bullet V^3$$

Dove:

- ρ = densità dell'aria (1,225 Kg/m^3);
- A = area trasversale della vena del vento che investe il generatore ($A= \Pi r^2$) in m^2;
- V = velocità del vento in m/s

Ma non tutta l'energia cinetica del vento può essere sottratta e trasformata in energia elettrica: il rendimento massimo teorico è del 59,3%, il rendimento pratico tra il 25% ed il 35% in considerazione di attriti, imperfezioni tecniche e perdite elettriche.
L'esperienza conferma che la potenza realmente producibile da un generatore eolico si può esprimere con la formula:

$$P = 0,15 \bullet D^2 \bullet V^3$$

Mappa complessiva della velocità media del vento a 50m s.l.t.

Mappa elaborata dal CESI in collaborazione con il Dipartimento di Fisica dell' Università di Genova nell'ambito della Ricerca di Sistema.

m/s
3 4 5 6 7 8 9 10 11

Figura 1 *Mappa eolica dell'Italia*

dove D = diametro del rotore espresso in metri. Per un generatore del diametro di 2,4 metri la potenza ricavabile, con un vento di 8 m/s, sarà:

$$P = 0{,}15 \bullet 2{,}4^2 \bullet 8^3 = 442 \text{ W}$$

Poiché la potenza è proporzionale al cubo della velocità, si può verificare facilmente che essa raddoppia passando da 8 m/s a 10 m/s.

LA NOSTRA SITUAZIONE REGIONALE

La Puglia, regione in cui noi viviamo, presenta un clima favorevole sia per il vento che per il sole: la ventosità media risulta tra 5 e 7 m/s con produzione annua di 1200 KWh per KW installato, mentre la produzione annua di un sistema fotovoltaico è di 1500 KWh per ogni KW di picco installato. Non a caso in Puglia esiste il più importante produttore italiano di grandi generatori eolici (IWT di Taranto) ed uno dei pochissimi produttori italiani di pannelli fotovoltaici (DEPASOL di Trinitapoli). In Italia la Puglia dà un grande contributo nella produzione di energia "verde" da fonte eolica (243 MW su 1135), con previsione di installazione di altri 130 MW entro il 2005.

POSSIAMO NOI MISURARE IL VENTO?

Esistono metodi empirici per misurare il vento: il più noto è quello proposto dall'ammiraglio Beaufort fondato sull'osservazione di alcuni caratteri indicativi della natura. La **scala di Beaufort** è di solito utilizzata per indicare la forza del mare, ma con essa è possibile valutare il vento a terra convertendo i caratteri indicativi come indicato nella tabella 1.

I venti di forza 0, 1 e 2 sono insufficienti a produrre energia eolica. Sono utilizzabili i venti di forza 3, 4, 5 e 6. Con venti da forza 7 in su, devono intervenire dei meccanismi di autoprotezione dei generatori.
Per chi volesse invece misurare il vento con precisione, è possibile comprare un anemometro già pronto oppure reperire sulle riviste progetti interessanti per realizzarlo.
Noi abbiamo scoperto su INTERNET una interessante soluzione.
Il sito *www.alphalink.com.au/~derekw/ane/anemain.htm* propone a meno di 100 euro un kit che misura velocità e direzione del vento, memorizza i dati e li scarica in un foglio elettronico.

SCELTE DI PROGETTO

Prima di iniziare la costruzione di un generatore eolico occorre fare alcune scelte di progetto in merito alla sua destinazione, al suo dimensionamento ed alla sua installazione.

Destinazione

Il generatore deve essere adatto ad alimentare una casa isolata presenziata con continuità solo nei mesi estivi. Il sistema sarà di tipo isolato (stand-alone) e necessiterà di accumulatori per-

Forza del vento	Velocità in m/s	Definizione	Caratteri indicativi a terra
0	0	Calma	Calma. Il fumo sale verticalmente
1	0 – 1,5	Bava di vento	Il vento piega il fumo
2	2 - 3	Brezza leggera	Le foglie stormiscono
3	3 – 5,5	Brezza tesa	Le foglie e i piccoli rami si muovono
4	6 - 8	Vento moderato	Si muovono i rami, si sollevano la carta e la polvere
5	8,5 - 10	Vento teso	Cominciano ad oscillare i piccoli alberi
6	11 - 14	Vento fresco	Si muovono i grossi rami, i fili metallici sibilano
7	14,5 - 17	Vento forte	Si muovono i grossi alberi, difficoltà a camminare controvento
8	17,5 - 20	Burrasca moderata	Si rompono i rami degli alberi
9	21 - 24	Burrasca forte	Oggetti e tegole asportati, i fabbricati possono subire qualche danno
10	24,5 - 28	Burrasca fortissima	Alberi abbattuti o sradicati, notevoli danni ai fabbricati
11	29 - 32	Fortunale	Devastazioni gravi
12	Oltre 33	Uragano	Devastazioni gravissime

Tabella 1: Scala di Beaufort

ché la produzione dell'energia eolica non coincide con la richiesta.

Potenza

La potenza nominale del generatore (calcolata con velocità del vento di 9 m/s) sarà di 500 W. Secondo i dati di esperienze simili, l'energia prodotta con questo generatore a velocità media di 5 m/s è di 50 KWh al mese.

Corrente continua o alternata?

Dovendo caricare degli accumulatori, ci servirà una corrente continua. Il generatore che costruiremo è un alternatore e quindi, per avere la corrente continua, occorrerà dotarlo di raddrizzatore.

12 o 24 Volt?

Un sistema che carica un accumulatore a 12 V presenta il vantaggio di poter utilizzare gli apparecchi elettrici creati per l'automobile. Allo stesso tempo presenta lo svantaggio di dover operare con correnti molto alte (una potenza di 1000 W richiederebbe una corrente di circa 100 A). Correnti alte richiedono grossi cavi e producono notevoli perdite di potenza. Un buon compromesso è quello di utilizzare un sistema

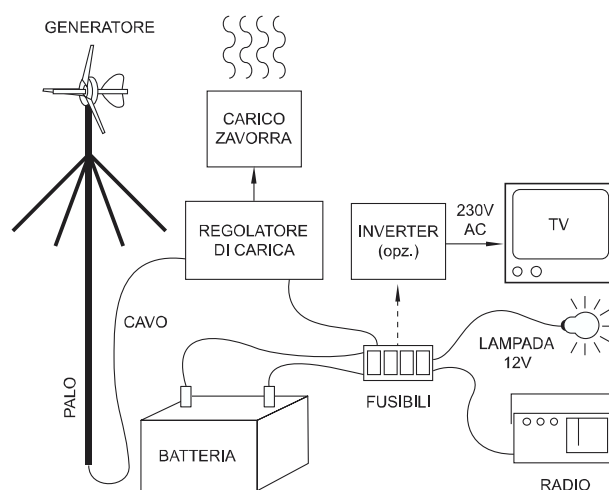



Figura 2 *Elementi di un sistema di generazione a energia eolica*

che carica gli accumulatori a 24 V.

ELEMENTI DI UN GENERATORE EOLICO

Un generatore eolico (figura 2) è costituito da diversi sottosistemi che qui elenchiamo:

- **Sistema per convertire la velocità del vento in energia meccanica di rotazione**: generalmente è un "mulino" a tre pale, ma ci sono tipologie diversissime e fantasiose.
- **Sistema per convertire l'energia meccanica di rotazione in energia elettrica**: in pratica è un generatore di energia elettrica mosso dalle pale, che per impianti di piccola taglia deve avere una uscita in corrente continua per poter caricare gli accumulatori.
- **Sistema per orientare il generatore rispetto al vento**: per i piccoli sistemi l'orientamento migliore rispetto al vento si raggiunge in maniera passiva per mezzo della "coda".
- **Sistema di protezione contro le velocità eccessive**: è prudente prevedere un sistema di protezione del generatore in caso di vento molto forte, ad esempio un automatismo che cambia il piano delle pale rispetto al vento.
- **Sostegno per posizionare il generatore ad altezza giusta**: più il generatore è in alto e più produce. Si usano in genere pali lunghi da 10 a 20 metri, realizzati con tubi per acquedotto da 2 o 3 pollici.
- **Sistema di carica degli accumulatori con protezione dalla sovraccarica**: gli accumulatori devono essere protetti contro la carica eccessiva deviando la corrente in eccesso su un carico "zavorra".
- **Conversione da tensione continua a tensione alternata**: per alimentare con gli accumulatori la maggior parte degli apparecchi elettrici, occorre un convertitore dalla tensione continua a quella alternata di 230 V (INVERTER).

CHE COS'È UN SISTEMA DI GRID-CONNECTED?

Chi ha un generatore di energia "verde" può accedere ai finanziamenti pubblici tramite le Regioni, purché si allacci alla rete elettrica. In questo modo cede alla rete l'energia prodotta in eccesso e prende energia da essa quando l'autoproduzione è insufficiente o mancante. Il possessore del generatore paga solo la differenza tra l'energia presa e quella ceduta, misurata con un sistema a due contatori. Se l'autoproduttore dovesse risultare in credito, riceverebbe un "buono" per i prossimi consumi.
Il sistema grid-connected, allettante per il contributo pubblico, si rivela scoraggiante per altri aspetti: la lunghezza della burocrazia, l'obbligo di affidamento dei lavori ad operatori riconosciuti e l'uso di apparecchiature omologate (tra cui dei costosi inverter sincronizzati con la rete).
Il sistema grid-connected è stato introdotto in Italia per poter connettere in rete i generatori fotovoltaici dei privati con taglia fino a 20 KW in adempimento della Direttiva Europea sui "10.000 tetti fotovoltaici". Con il D.Lgs. 387/03, la possibilità di connessione in rete è stata concessa anche ai generatori eolici di piccola taglia.

I GENERATORI EOLICI E GLI AMBIENTALISTI

I generatori eolici sono amati ed odiati dagli ambientalisti.
L'energia eolica è senza dubbio una forma ideale di energia, perché è rinnovabile e pulita. Per ogni Kilowattora di energia prodotta infatti si evita l'emissione di:

- 1000 grammi di anidride carbonica (CO_2)
- 1,4 grammi di anidride solforosa SO_2
- 1,9 grammi di ossidi di azoto (SOx)

Si stima che i generatori eolici oggi installati in Italia evitino ogni anno l'emissione nell'atmosfera di 1,8 milioni di tonnellate di CO_2.
I movimenti ambientalisti osservano che i grossi generatori eolici alterano il profilo naturale del territorio, sono un pericolo per gli uccelli, generano rumore e sottraggono terreno all'agricoltura per far posto a strade e

linee elettriche.
La discussione sembra lunga e senza via di uscita: ma intanto, mentre l'Italia ha installato fin'ora 1135 MW di generatori eolici, la Germania ne ha installati 15000, la Spagna 7000 e la piccola Danimarca 3000.

Coordinatori progetto Eolo: proff. Antonio Cecere e Beatrice Paparella

DOCUMENTAZIONE SULLA PRIMA PARTE

Libri e guide

Paul Gipe, "Elettricità dal vento. Impianti di piccola scala"
F. Muzzio Editore
Pirazzi-Bigotti – "Le vie del vento. Tecnica, economia e prospettive del mercato dell'energia eolica"
F. Muzzio Editore
"Energia elettrica dal vento" – Guida ADICONSUM (scaricabile da INTERNET)

Siti INTERNET

www.arturo.derisi.unile.it/Energie_rinnovabili/dowloads/ER_lez_6.pdf
www.scoraigwind.com/download/Bookletwind.pdf
www.etaflorence.it/pdfs/siwern_2004/enea%20pirazzi.pdf
www.homepower.com/files/SagrilloA&OTech.App.pdf
www.energia-eolica.it

Le so tutte!!!

Risposta al quesito codice LST23803

Chiave:
SCATTERNET

Il vincitore di Marzo 2005 (LST23702)
I nostri complimenti a **Prist Mariorosario** di Grottammare (AP)
che vince un abbonamento a Fare Elettronica!

Intelligenza artificiale

Nella scorsa puntata sono state introdotte le reti neurali, il loro funzionamento ed alcune delle loro caratteristiche. In questo articolo invece analizzeremo più in dettaglio alcuni aspetti pratici legati alla loro implementazione ed al loro utilizzo. Per fare questo ci serviremo di un paio di esempi pratici.

INTRODUZIONE

Per completare e mettere in pratica le conoscenze acquisite nello scorso articolo vedremo ora alcuni esempi di utilizzo delle reti neurali. I due esempi scelti mostrano abbastanza bene sia la tipologia delle possibili applicazioni delle reti neurali (e le loro potenzialità), sia le modalità e le procedure per utilizzarle. Partendo dagli esempi ci soffermeremo sui dettagli relativi all'implementazione e alle procedure di addestramento delle reti. Saranno presi in considerazione due esempi, uno basato su una semplice rete di percettroni, e l'altro su un percettrone multistrato (*Multi Layer Perceptron, MLP*), per il quale sarà utilizzato l'algoritmo di backpropagation.
Dal momento che nei seguenti paragrafi si farà spesso riferimento ad alcuni concetti già illustrati nello scorso articolo, consiglio vivamente a chi non l'avesse già fatto di (ri)leggerlo.

ESEMPIO 1: RICONOSCERE I COLORI

Immaginate che in una particolare applicazione sia necessario riconoscere determinati colori, ed in particolare alcuni colori descritti in maniera abbastanza imprecisa come celeste, indaco, viola, marrone, verdino, arancione, e grigio… Il colore deve essere riconosciuto a partire dai segnali forniti da tre sensori ottici (fotodiodi) che rilevano la quantità dei tre colori primari: rosso, verde e blu (in sigla RGB). Risolvere questo problema con i metodi tradizionali non è difficile, ma può dare risultati non pienamente soddisfacenti a causa dell'imprecisione insita nel problema stesso. L'impiego di una rete neurale invece può essere una buona soluzione, dal memento che permette di trasferire all'hardware delle conoscenze soggettive difficilmente modellizzabili matematicamente. Visto il

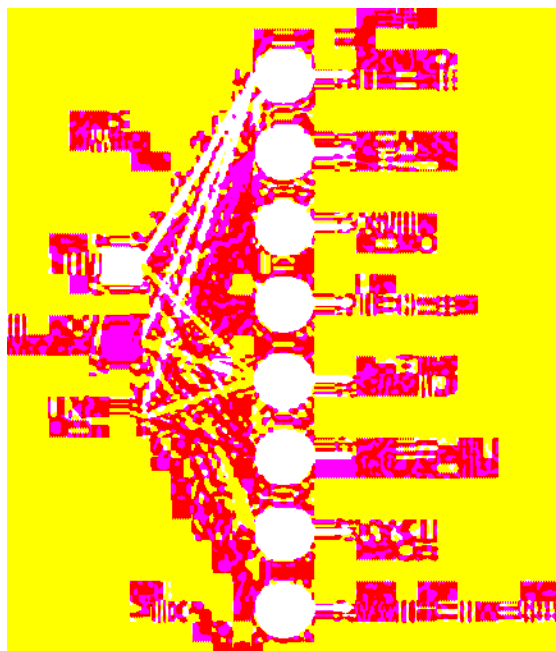
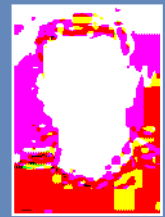



Figura 1 *Rete neurale utilizzata per riconoscere i colori*

di Antonio Di Stefano
a.distefano@farelettronica.com

Implementazione delle reti neurali

tipo di problema, si potrebbe utilizzare una rete di percettroni, dotata di tre ingressi, e di tante uscite quanti sono i colori da riconoscere (figura 1).
I tre ingressi della rete ricevono i valori (sotto forma di campioni numerici) delle componenti rosso, verde e blu provenienti dai sensori, le uscite invece segnalano con la loro attivazione il colore eventualmente riconosciuto.

Vista l'imprecisione che caratterizza il problema, è il caso di utilizzare un funzione di attivazione a valori continui, ad esempio quella lineare a tratti. Per addestrare la rete sarà necessario presentarle un insieme di colori scelti a caso, e per ciascuno indicare la risposta voluta. Da notare che in questo modo la rete eredita in un certo senso le conoscenze soggettive del suo addestratore, e tendenzialmente risponderà nello stesso modo (cioè ad esempio avrà la stessa idea di "indaco").
L'addestramento della rete, nonostante non presenti particolari difficoltà, potrebbe essere fatto "off-line" su PC. Una volta ottenuti i pesi definitivi, si potrebbe implementare la rete per il solo funzionamento "in avanti", anche su sistemi molto più piccoli e semplici, come ad esempio quelli a microcontrollore. Vedremo come sia possibile fare questo nel prossimo paragrafo.

ASPETTI IMPLEMENTATIVI

Iniziamo subito col fare una prima distinzione: implementare una rete neurale utilizzando un linguaggio ad alto livello (C, LISP, Basic…) su un sistema di calcolo sufficientemente potente, è molto diverso dall'implementarla in assembler su un sistema dotato di un piccolo processore (al limite un microcontrollore) e con risorse di memoria molto limitate. Nel primo caso tipicamente si possono utilizzare strutture dati complesse per descrivere la rete, i singoli strati, ed i neuroni. Inoltre è possibile utilizzare numeri in virgola mobile per rappresentare le grandezze caratteristiche della rete (gli ingressi, le uscite, i pesi, le attivazioni…), e funzioni matematiche abbastanza complesse.
Nel secondo caso invece occorre procedere con molta più attenzione e curare maggiormente diversi aspetti del programma che possono non essere del tutto intuitivi. Prendiamo in considerazione questo secondo caso, utilizzando come modello l'esempio precedente.

Rappresentazione della rete

Abbiamo visto nello scorso articolo quali sono le parti che costituiscono una rete neurale. Per rappresentare la rete occorre memorizzare tutte queste informazioni, ed in particolare quelle relative agli ingressi, ai pesi, alle attivazioni, ed alle uscite. In genere sarebbe necessario memorizzare anche qualche informazione sulla topologia della rete, come ad esempio il numero di ingressi, di uscite, di strati, e di neuroni per strato. Queste informazioni sono spesso codificate implicitamente nello stesso programma *(nei loop)* quando la topologia della rete è fissa e regolare. Può essere conveniente organizzare le informazioni in un area di memoria ben definita, in modo da potere accedere in maniera semplice e veloce ai dati che servono. In particolare i pesi, che sono i dati su cui occorre lavorare di più, possono essere memorizzati in un'area contigua ed in maniera consecutiva, in modo da potervi accedere utilizzando degli indici e delle costanti (figura 2). Nel nostro esempio ci saranno 3 ingressi collegati ad 8 neuroni, considerando anche l'ingresso di bias occorreranno quindi in totale 4x8 (32) "aree" di memoria contigue. In questo modo l'indirizzo del peso relativo al collegamento tra l'ingresso i ed il neurone j sarà dato dalla formula p = i + 4*j + area_pesi (la

formula vale se i e j sono numerati a partire da 0, in caso contrario basta sottrarre 1 da entrambi).

Funzione di attivazione

Come calcolare il valore della funzione di attivazione? Se si vuole utilizzare una funzione a gradino non ci sono particolari problemi. Può essere un po' più difficile invece utilizzare la funzione sigmoide, soprattutto se non si sta utilizzando un linguaggio ad alto livello ed un processore sufficientemente potente. In questo caso può essere una buona soluzione utilizzare una *lookup table*, cioè una tabella (un array monodimensionale) che riporti il valore che la funzione assume in alcuni punti (figura 3a). Il numero di punti da memorizzare dipende essenzialmente dalla disponibilità di memoria che si ha. Questo metodo permette di calcolare molto velocemente il valore approssimato della funzione, è necessario infatti soltanto un'addizione ed un accesso in memoria.

Per sistemi di piccole dimensioni può essere una buona scelta quella di utilizzare una funzione di attivazione lineare. In questo caso basta utilizzare come valore di attivazione quello di ingresso se il valore è inferiore ad una certa soglia, o il valore della soglia stessa in caso contrario (si esegue in pratica un *clipping*), come mostrato in figura 3b.

Figura 2 Esempio di organizzazione dei pesi in memoria

Rappresentazione delle grandezze

Nella scorsa puntata abbiamo supposto che i vari segnali (ingressi, uscite, attivazioni, pesi…) fossero dei numeri reali.

Tuttavia la gestione di una rappresentazione in virgola mobile può risultare alquanto pesante se non è supportata direttamente dall'hardware. Per fortuna, con qualche accorgimento è possibile utilizzare anche i comuni numeri interi (a virgola fissa, in complemento a 2) per rappresentare le grandezze che ci interessano. Descrivere in dettaglio le tecniche che si usano in questi casi sarebbe un po' troppo lungo, comunque l'idea di base è quella di approssimare i numeri e/o prevenire gli overflow troncando opportunamente i risultati ottenuti. Per fare questo è sufficiente eseguire degli shift aritmetici a destra (che equivalgono a delle divisioni per potenze di 2) quando si eseguono determinate operazioni. Inoltre occorre in genere utilizzare un numero di bit maggiori per calcolare i risultati intermedi, e solo dopo si eseguono gli shift.

Figura 3 Funzioni di attivazione a) sigmoide con look-up table, b) lineare a tratti

Rimando comunque ad uno dei riferimenti bibliografici per maggiori dettagli.
Ritornando all'esempio, tutte le grandezze potrebbero essere rappresentate da singoli byte. In questo modo l'occupazione di memoria sarebbe di circa 3+32+8 = 43 byte! A questi vanno aggiunti ovviamente quelli utilizzati come variabili temporanee per eseguire i calcoli.

Condizionamento dei dati

I dati in ingresso ad una rete neurale devono essere di solito opportunamente "condizionati", cioè è necessario eseguire delle elaborazioni preliminari prima di poterli utilizzare come ingressi. Il caso più comune è quello della riscalatura, che consiste nel riportare l'intervallo dei dati originali a quello previsto dagli ingressi della rete. Nel caso dell'esempio, dal momento che abbiamo supposto di utilizzare dati a 8 bit con segno (valori da –128 a 127), e presumibilmente i sensori forniranno un dato ad 8 bit senza segno (valori da 0 a 255), sarà necessario dividere per due i dati in ingresso, riportandoli nell'intervallo da 0 a 127. A volte, soprattutto quando si utilizzano reti neurali con rappresentazione floating point è necessario fornire ingressi *normalizzati* (compresi tra 0 ed 1), è sufficiente in questi casi dividere i dati per il valore massimo che essi possono assumere. Un altro esempio di condizionamento è quello che bisogna utilizzare quando come valore d'ingresso sono richieste grandezze come "media su un intervallo di tempo", "derivata", "massimo", "minimo", e simili. In questo caso i valori devono essere calcolati a partire dai dati, prima di poterli fornire alla rete neurale. In alcuni casi simili considerazioni valgono anche per le uscite.

In conclusione il programma deve eseguire i seguenti passi per far funzionare la rete in avanti:

1. leggere gli ingressi, condizionarli e memorizzarli.
2. Calcolare per ogni neurone:
3. la somma pesata degli ingressi:
   $I_n = W_{1n}*i_1 + W_{2n}*i_2 + W_{3n}*i_3 + W_{bn}*1$
4. il valore della funzione di attivazione:
   $a_n = g(I_n)$ [e memorizzarla]

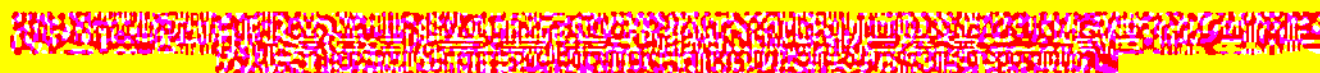
5. Se necessario scegliere come uscita valida quella con il valore più grande

Le formule riportate sopra si riferiscono ad un generico neurone (l'n-esmino), mentre il termine W_{bn}*1 è quello relativo al fattore di bias.
I prodotti tra i pesi e gli ingressi possono essere ottenuti con una moltiplicazione a 8x8 bit con segno, ottenendo un risultato a 16 bit. Per riportare il risultato ad 8 bit (approssimandolo) è sufficiente eseguire uno scorrimento a sinistra di un bit (per cancellare il bit di segno "in più") e considerare come risultato soltanto gli 8 bit più significativi. Per sommare i 4 termini invece è necessario utilizzare un accumulatore a 16 bit e scorrere il risultato finale a destra di due bit (divisione per 4) per riportare il valore ad 8 bit, oppure applicare direttamente il clipping della funzione di attivazione al numero a 16 bit, ottenendo così il valore dell'uscita. Ora la nostra prima semplice rete neurale dovrebbe essere pronta e funzionante.

ADDESTRAMENTO

Ritorniamo ancora sulla fase di addestramento, per analizzare meglio alcune caratteristiche e procedure. Come già detto per addestrare una rete neurale si utilizza un training set composto da alcuni esempi (ingressi) e dalle risposte volute per quegli ingressi. Un requisito importante per il *training set* è che esso deve essere il più rappresentativo possibile, cioè deve contenere tutti gli ingressi che si vogliono far riconoscere alla rete, ed anche diversi casi estranei o "ambigui", per i quali verrà mostrata la risposta voluta. Purtroppo non esistono teorie precise riguardo al numero ottimale di esempi da utilizzare, per cui di solito ci affida più che altro all'esperienza. Abbiamo già visto che il procedimento è il seguente: 1) si inizializzano i pesi in maniera casuale, 2) si applica un ingresso di esempio alla rete e la si fa funzionare in avanti, 3) si calcola l'errore, cioè la differenza tra l'uscita della rete e quella voluta, 4) si aggiornano i pesi con una regola dipendente dal tipo di rete 5) si ripete dal punto 2) un numero fissato di volte, o fino a quando l'errore medio della rete non diminuisce sotto una soglia stabilita. Giusto per fare un esempio numerico, nel caso del riconoscitore di colori un esempio del training set potrebbe essere il seguente: ingresso {117, 192, 239} (valori RGB corrispondenti al celeste), uscita voluta {127, 0, 0, 0, 0, 0, 0, 0} (uscita "celeste" attiva al massimo, altre inattive). Un aspetto importante che di solito non viene evidenziato abbastanza è che gli esempi devono essere mostrati alla rete in ordine casuale, e grosso modo un numero di volte uguale per tutti (un ciclo in cui vengono mostrati tutti gli esempi viene chiamato "epoca"). Se si utilizza una sequenza fissa o comunque regolare, la rete potrebbe non imparare completamente! Questo dipende dal fatto che la convergenza verso il minimo dell'errore è ottenuta in modo iterativo (con un algoritmo simile a quello del gradiente), che può rimanere "bloccato" se si forniscono degli stimoli in qualche modo periodici. Uno strumento utile per controllare che la rete stia effettivamente imparando è il grafico che riporta l'errore medio (su tutte le uscite) al variare dei cicli di addestramento.

Figura 4 Esempio di grafico dell'errore di una rete neurale

Questo grafico deve mostrare una tendenza a decrescere dopo un certo numero di iterazioni (figura 4).
Un altro suggerimento utile è quello di preparare un secondo training set, differente dal primo, che servirà per testare la rete, una volta completato l'addestramento.

ALGORITMO DI BACK-PROPAGATION

Prima di passare al prossimo esempio, in cui utilizzeremo il percettrone multistrato mostrato in figura 5, è il caso di dare un'occhiata all'algoritmo utilizzato per addestrare questo tipo di reti. Come detto nello scorso articolo la difficoltà maggiore risiede nel fatto che è possibile calcolare l'errore solamente per l'ultimo strato, ma non per quelli intermedi (nascosti). Questa difficoltà è superata con l'algoritmo di back-propagation. Il procedimento è il seguente:

1. Si ricava per ogni uscita un termine proporzionale all'errore:
$$\Delta i = Err i * g'(I i)$$
Dove Err*i* è l'errore all'i-esima uscita (Targeti-Uscitai), e g'(I*i*) è la derivata della funzione di attivazione calcolata con la somma pesata degli ingressi del neurone i-esimo.
2. Si aggiornano i pesi dell'ultimo strato con la seguente formula:
$$W_{ij} = W_{ij} + \alpha * a_j * \Delta_i$$
Dove α è il consueto coefficiente di learning rate (valori comuni sono compresi tra 0,01 e 1), a_j è l'attivazione del j-esimo neurone dello strato precedente.
3. Per ogni strato intermedio si ripartisce l'errore in base alla somma dei pesi di ogni neurone:
$$\Delta_j = g'(I_j) * \sum (W_{ij} * \Delta_i)$$
4. Si aggiornano i pesi dei collegamenti con lo strato precedente:
$$W_{kj} = W_{kj} + \alpha * a_k * \Delta_j$$
5. Si ripetono gli ultimi due passi per tutti gli strati, fino ad arrivare a quello di ingresso, in cui il termine a_k è sostituito dal valore dell'ingresso k-esimo (i_k).

La derivata della funzione di trasferimento può essere calcolata in diversi modi: nel caso della funzione sigmoide essa è funzione del valore assunto dalla funzione stessa, quindi si può usare il valore precedentemente calcolato, o in ogni caso si può ricorrere ad una *look-up table*. Nel caso della funzione lineare a tratti, essa vale 1 nel tratto inclinato, e 0 nei tratti piani. Per avere un comportamento migliore dell'algoritmo si può utilizzare un piccolo valore positivo al posto dello 0 (moltiplicare per un valore frazionario positivo equivale in genere ad eseguire degli shift aritmetici a destra).

L'intero procedimento può sembrare un po' complicato, ma una volta compreso il meccanismo non presenta particolari difficoltà…

ESEMPIO 2: RICERCARE OGGETTI

Consideriamo adesso un esempio un po' più complesso ma molto più interessante, perché fa apprezzare meglio le potenzialità delle reti neurali. Immaginiamo di dovere controllare gli spostamenti di un piccolo robot che deve esplorare un ambiente chiuso alla ricerca di alcuni piccoli oggetti. Supponiamo che sia gli oggetti, sia le pareti riflettano la luce molto più del pavimento (oppure che siano semplicemente di due colori diversi, e distinguibili usando un filtro colorato…). Il robot è dotato di una piccola telecamera posta sulla parte anteriore. Il problema consiste nel ricavare la direzione degli spostamenti (avanti, indietro, destra, sinistra) in base all'immagine fornita dalla telecamera. In particolare se nel campo visivo è presente uno dei piccoli oggetti,

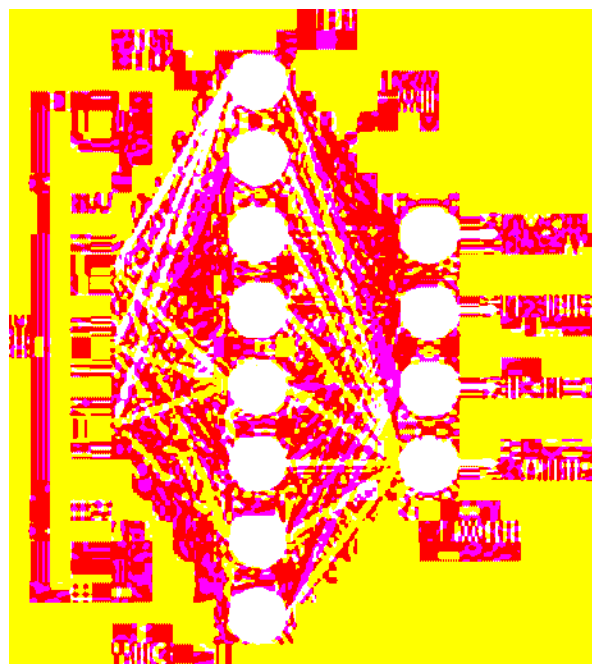



Figura 5 *Percettrone multistrato utilizzato nell'esempio 2*

il robot deve raggiungerlo (questa condizione equivale ad avere l'oggetto in basso al centro nell'immagine), se sono presenti delle pareti (grosso oggetto di colore uniforme nella parte alta, destra o sinistra) il robot deve andare in direzione opposta. Quando il robot raggiunge un oggetto entrerà in funzione un meccanismo apposito per raccoglierlo (togliendolo così dal campo visivo). Se il robot non riesce a vedere oggetti nel suo campo visivo deve procedere in avanti (cambierà direzione quando incontrerà una parete).
Risolvere questo problema con metodi tradizionali non è affatto banale! Occorrerebbe in ogni caso scrivere moltissime linee di codice… Vediamo invece come una rete neurale (eventualmente aiutata da routine di controllo di tipo tradizionale) può offrire una soluzione semplice ed efficace.

Figura 6 *Esempio di training set utilizzato per l'esempio 2*

Possiamo supporre di utilizzare una rete che prenda in ingresso direttamente l'immagine della telecamera, e dia in uscita direttamente la direzione consigliata. L'ingresso potrebbe essere una matrice di 4x3 pixel ottenuta dall'immagine riscalata ed in scala di grigio (una matrice un po' più grande sarebbe più realistica…), e l'uscita 4 segnali che indicano le direzioni "avanti", "indietro", "destra", "sinistra" (che poi verranno elaborate opportunamente per controllare i motori). Vista la complessità del compito richiesto alla nostra rete, è bene utilizzare un MLP di almeno 3 layer (figura 4): 12 neuroni di ingresso (la matrice 4x3), 8 neuroni nascosti, 4 neuroni di uscita (le direzioni). Come già detto non esiste nessuna giustificazione teorica a questa scelta, essa deriva più che altro da un *trade-off* tra quantità di memoria richiesta e capacità della rete, in ogni caso la scelta non è critica. Il numero di pesi richiesti è 12*8 + 8*4 = 128 se non si utilizzano i termini di bias, e di 13*8 + 9*4 = 140 se si decide di utilizzarli. Considerando anche gli ingressi e le attivazioni del layer nascosto è necessario memorizzare da 152 a 164 variabili (che possiamo immaginare essere byte). La mappa di memoria per memorizzare questi dati può essere simile a quella utilizzata nell'esempio precedente, però in questo caso sono necessarie due aree per i pesi (W_{kj} e W_{ji}) e una terza area per le attivazioni del layer nascosto (a_j). Può essere una buona idea anche in questo caso utilizzare una funzione di attivazione lineare a tratti.
L'addestramento della rete può essere eseguito off-line su PC, utilizzando magari proprio le immagini catturate dalla telecamera nelle varie situazioni (figura 6). Oppure, se si decide di implementare l'algoritmo di addestramento sul robot, è possibile perfino insegnargli il suo compito guidandolo manualmente all'inizio! In ogni caso è importante che il training set comprenda tutte le situazioni in cui il robot si può trovare. Per addestrare la rete, trattandosi si un MLP, occorre utilizzare l'algoritmo di back-propagation.

Un aspetto molto interessante di questo approccio è che utilizzando un training set diverso è possibile insegnare al robot a comportarsi in maniera completamente diversa! Ad esempio potrebbe imparare senza difficoltà a seguire delle linee, a percorrere delle strade mantenendosi lontano dagli ostacoli, ad inseguire degli oggetti in movimento… E tutto questo senza la necessità di modificare una sola linea di codice!

LIMITI DELLE RETI NEURALI

A volte la prima impressione che si ha delle reti neurali è che esse possano essere la soluzione ideale per qualsiasi problema. Purtroppo non è così! Una rete neurale infatti può risolvere un determinato problema solo se esiste una relazione (descrivibile matematicamente) tra gli ingressi e le uscite. Questo è il caso degli esempi riportati. La rete in questi casi non fa altro che scoprire e imparare una funzione matematica "nascosta" (anche molto complessa) che mette in relazione gli ingressi con le uscite. Non è possibile ad esempio fare "prevedere" ad una rete neurale il valore futuro di titoli azionari soltanto in base ai loro valori passati, semplicemente perché il loro andamento è determinato in gran parte da altri fattori esterni (non sempre conoscibili). A volte invece alcuni problemi, pur se trattabili possono essere mal posti, e quindi non si ottengono risultati soddisfacenti.
Questo può succedere se non si forniscono alla rete tutte le informazioni necessarie per ricavare le conclusioni corrette, o li si fornisce in maniera errata. Un buon metodo per capire se ci si trova in uno di questi casi è quello di "immedesimarsi" nella rete neurale, cioè chiedersi se partendo di dati forniti in ingresso si sarebbe in grado di dare delle risposte corrette.
Inoltre non bisogna dimenticare che anche reti ben addestrate a volte possono dare dei risultati completamente imprevedibili in presenza di particolari ingressi, per questo motivo è sempre bene eseguire qualche controllo "di sicurezza", soprattutto in applicazioni particolarmente delicate o senza una supervisione umana.

CONCLUSIONE

La teoria alla base delle reti neurali non è certo semplice, ed è abbastanza difficile trattare l'argomento in modo semplice ed esaurire in un paio di articoli! Su Internet comunque è possibile trovare ulteriori informazioni, molti esempi di implementazione di reti neurali in diversi linguaggi, molti esempi applicativi, e anche programmi già pronti per l'addestramento e la simulazione di reti neurali su PC. In bibliografia sono riportati alcuni link estremamente utili da questo punto di vista. Studiando i codici e tenendo presente quanto detto è possibile acquisire una certa dimestichezza nell'uso di questo potente e versatile strumento. Il prossimo articolo di questa serie dedicata all'Intelligenza Artificiale riguarderà la Logica Fuzzy, un interessante metodo per implementare sistemi intelligenti in maniera semplice ed intuitiva.

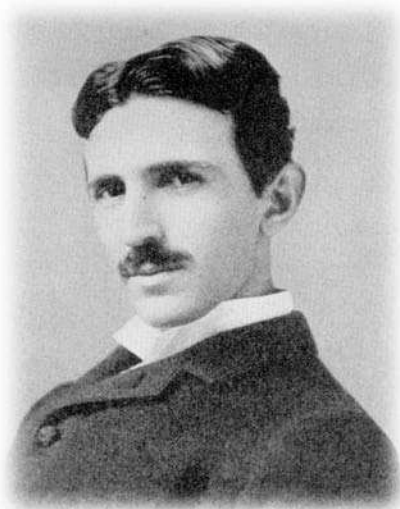
Bibliografia

1. Robert Gordon, "A Calculated Look at Fixed-Point Arithmetic", Embedded.com: http://www.embedded.com/98/9804fe2.htm
2. Karsten Kutza, "Neural Networks At Your Fingertips", sito Internet: http://www.neural-networks-at-your-fingertips.com/
3. Sito Internet: http://neuralnetworks.ai-depot.com/
4. Sito Internet: http://www.generation5.org
5. Sito Internet: http://sourceforge.net/

Dal 15 Luglio
in tutte le edicole

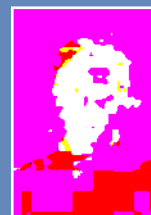
da NON PERDERE!!

Nikola Tesla

Brillante inventore e ingegnere elettrotecnico, Nikola Tesla ideò il sistema di produzione di corrente alternata di potenza, che consentì la distribuzione dell'energia elettrica a basso costo per gli impianti industriali e per le abitazioni civili. A Tesla è riconosciuta la paternità di oltre 100 brevetti negli Stati Uniti.

Nikola Tesla nacque a Smiljan, villaggio situato in Croazia, regione appartenente all'Impero Austro-Ungarico, il 9 luglio del 1856.
Il padre, il reverendo Milutin Tesla, era un sacerdote ortodosso serbo. La madre, Djuka Mandich, pur non avendo potuto frequentare la scuola, era una donna abile e molto intelligente. Di salute cagionevole fin da bambino, il giovane Nikola si mise presto in evidenza come studente brillante, dotato di un'ottima memoria fotografica. I genitori erano entrambi originari della Serbia occidentale. Nella famiglia,da generazioni, i figli maschi venivano forzatamente avviati a seguire la carriera ecclesiastica o quella militare, mentre, per le figlie femmine, il destino più ambito era quello di divenire mogli devote di ministri del culto o di ufficiali della milizia. Quindi, Nikola dovette imporsi e, contrastando il volere del padre, scelse la carriera di ingegnere elettrotecnico.

Animato da sogni ambiziosi e dotato di discrete capacità di scrittore, Tesla aveva un carattere molto disciplinato e in diverse occasioni dimostrò di avere una precisione quasi maniacale.
Avviatosi agli studi di ingegneria, egli frequentò dapprima l'università austriaca di Graz e in seguito quella di Praga (periodo dal 1879 al 1880). Durante il corso di studi a Graz, Nikola potè vedere la **dinamo di Gramme.** La macchina (figura 2), costruita da Gramme su suggerimento dell'italiano Antonio Pacinotti, aveva la caratteristica di operare in maniera reversibile, cioè poteva funzionare sia da generatore di corrente che da motore.
In questa occasione, Tesla iniziò a elaborare il progetto di un generatore di potenza in corrente alternata, atto a produrre corrente elettrica in maniera più economica, rispetto alla dinamo, che operava in corrente continua.

Dopo la laurea, ottenuta all'università di Praga nel 1880, Nikola Tesla trovò impiego a Budapest, in un ufficio telegrafico governativo. In quel periodo diede forma alla sua prima invenzione, un **ripetitore telefonico.** Successivamente, Tesla concentrò la propria attenzione sul principio del campo magnetico rotante e si dedicò alla pro-

di Agostino Rolando
a.rolando@farelettronica.com

gettazione di un **motore a induzione.**

Nel 1882 l'inventore si trasferì a Parigi, per lavorare presso la *Compagnia Continentale Edison;* qui, mentre era assegnato ad un incarico a Strasburgo, nel 1883, ricavando un po' di tempo oltre l'orario di lavoro, costruì un prototipo del motore a induzione.

Figura 2 Dinamo di Gramme

L'anno successivo Tesla decise di imbarcarsi per l'America. All'età di 27 anni, lo possiamo immaginare affacciato al parapetto della nave, in vista del porto di New York, con pochi spiccioli in tasca e con i suoi componimenti poetici nella valigia, sicuro di sè e ricco di idee e di progetti per il futuro.

Thomas Alva Edison assunse il giovane ingegnere come assistente fin dal suo arrivo in America. Tuttavia, tra i due nacquero presto degli attriti, al punto che, nel 1886, Tesla dovette abbandonare il suo impiego.
Dopo un periodo di difficoltà, durante il quale Tesla inventò un sistema di illuminazione ad arco voltaico, riuscì a farsi finanziare una somma sufficiente per costruire un proprio laboratorio a New York (figura 3), dove la sua mente fertile potè spaziare liberamente.
Tra i suoi numerosi esperimenti meritano di essere menzionati gli studi sul trasferimento di energia elettrica mediante risonanza e diversi tipi di lampade per illuminazione.
Tesla diede dimostrazioni pubbliche dei suoi esperimenti nel suo laboratorio, come ad esempio accendere lampade fluorescenti facendo fluire deboli correnti alternate ad alta tensione attraverso il proprio corpo (figura 4).

Il cosiddetto "Tesla Coil", che inventò nel 1891, venne utilizzato negli apparecchi televisivi per produrre l'alta tensione necessaria per il funzionamento del tubo catodico.
A questo proposito, Tesla rimane famoso per i generatori ad alta tensione con i quali riusciva a stupire il pubblico, facendo scoccare scintille di decine di metri di lunghezza.
Oggigiorno si possono trovare interessanti studi, compiuti da sperimentatori australiani, su generatori ispirati al "Tesla Coil", che dimostrano come sia possibile produrre con tecniche "casalinghe" scintille della lunghezza di 3-4 metri, con una potenza assorbita di quasi un kilowatt (vedere figure 5 e 6).

Figura 3 Tesla nel suo laboratorio

Figura 4 Una dimostrazione di Tesla

Nel 1889 lo scienziato ottenne la cittadinanza americana. L'anno pre-

cedente aveva già brevettato il suo sistema di generazione di corrente alternata di potenza.
Il cuore di questo sistema era un efficiente generatore polifase che egli aveva sviluppato con propri mezzi.

Nel 1885, *George Westinghouse,* a capo della *Westinghouse Electric Company,* con sede a Pittsburgh, acquistò i diritti dell'invenzione e iniziò una massiccia campagna di propaganda del nuovo sistema di produzione di corrente elettrica. Questo diede luogo a un feroce scontro tra Edison, padrino del sistema di generazione in corrente continua e il duo Tesla-Westinghouse, il cui metodo vinse la competizione, soppiantando a poco a poco il metodo a corrente continua e divenendo infine il principale sistema di produzione e trasporto dell'energia elettrica negli USA.
Westinghouse utilizzò il sistema in AC per illuminare *l'Esposizione Mondiale di Chicago,* nel 1893. Il successo di questa installazione fu fondamentale per vincere il contratto di costruzione di una centrale elettrica alle cascate del Niagara. L'impianto fu in grado di produrre l'energia sufficiente ad illuminare la città di Buffalo.

Uomo brillante ed eccentrico, Tesla si dedicò alla progettazione di diversi tipi di turbine e si interessò anche alla trasmissione wireless di segnali di potenza. A Colorado Springs, dove egli soggiornò dal maggio del 1899 fino agli inizi del 1900, Tesla fece quella che considerò la sua più importante scoperta: le onde stazionarie terrestri. In base a questa teoria, egli ipotizzò che il nostro pianeta si sarebbe potuto utilizzare come conduttore e avrebbe avuto un picco di risposta a segnali elettrici di una determinata frequenza.
Ritornato a New York nel 1900, Tesla iniziò la costruzione di una torre per trasmissioni radio, a Long Island, supportato dal finanziere Pierpont Morgan, ma Il progetto fu poi abbandonato per sopraggiunti problemi di carattere finanziario.

Nel 1915 Tesla rimase molto deluso dalla mancata assegnazione del Premio Nobel, che avrebbe dovuto condividere con Edison. Ricevette tuttavia la *Edison Medal* nel 1917, onorificenza molto prestigiosa offertagli *dall'American Institute of Electrical Engineers.*

Tesla ebbe pochissimi amici, tra cui alcuni famosi scrittori come *Mark Twain* o *Francis Marion Crawford.* Tesla risultò completamente inadatto a gestire le proprie risorse finanziarie.
Una sua caratteristica era quella di stupire i reporters con eccezionali rivelazioni, come quella di avere ricevuto segnali radio da altri pianeti o come la sua affermazione di avere inventato un "raggio della morte" in grado di distruggere 10.000 aerei a 250 miglia di distanza.

Tesla morì a New York il 7 gennaio del 1943, con al suo attivo oltre un centinaio di brevetti.
In suo ricordo, a Belgrado fu eretto il museo a lui intitolato e il suo nome fu dato all'unità di misura del campo di induzione magnetica.

Figura 5 *Tesla Coil, apparecchiatura completa*

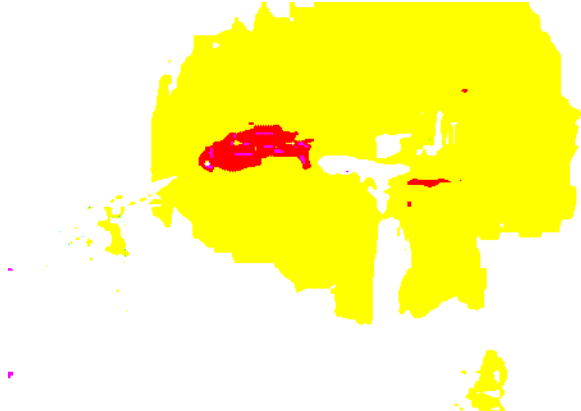



Figura 6 *Tesla coil in azione*

Stazioni meteorologiche

STAZIONE METEO WIRELESS

WS2300

WS3600

STAZIONE METEO WIRELESS con TOUCH SCREEN

Eppure

Per un tecnico elettronico l'arte di arrangiarsi è una attività normale anzi un dovere. A volte la ricerca di soluzioni porta ad ottenere un risultato nel breve tempo dell'idea e della sua realizzazione. Non avrei mai pensato di impiegare 20 anni e più per trovare una soluzione ad un problema che invece era li pronta a farsi cogliere senza alcuno sforzo.

20 anni fa, era tanta la voglia di costruire circuiti stampati che anche se dovevo passare gran parte del mio tempo libero ad incollare etichette trasferibili, il risultato era compensato da un buon lavoro.
Era veramente dura perché prima di tutto dovevo realizzare il master e poi con artifici più o meno convenzionali, lo riproducevo sulla basetta ramata con l'uso di trasferibili per elettronica. In pratica, riproducevo su foglio trasparente il master e poi , con un punzone, segnavo tutti i fori sulla basetta e poi, su questi fori posizionavo le piazzole e da queste, che usavo come riferimento, facevo partire le linee trasferibili per disegnare l'intero circuito stampato.
Dopo circa due giorni di lavoro avevo finalmente terminato ed alla fine, immergevo il tutto nella soluzione acida. Dopo pochi minuti il risultato era li, disponibile a funzionare, se non avevo dimenticato nulla ma generalmente una o due piste o una piazzola era saltata e se mi andava bene potevo risolvere con dei fili di rame e saldatore.
Che fatica! però vi lascio immaginare il livello di soddisfazione quando tutto funzionava.
Presto le difficoltà hanno superato il piacere di ottenere il risultato voluto, soprattutto perché non era pensabile realizzare una basetta più grande di sette otto centimetri per lato.
Allora sono passato direttamente al disegno dello stampato sulla superficie di rame, con l'ausilio di pennarelli indelebili che ben presto hanno dimostrato la loro inefficienza, soprattutto quando ci si trovava di fronte ad errori e correzioni e comunque a fronte di una grande fatica.
Più tardi, in anni quasi recenti, sono passato alla realizzazione dei prototipi utilizzando le basette pre-forate che senza dubbio saltavano tutto il processo della realizzazione su rame, ma che comportavano una assoluta attenzione nella fase di produzione.
Tutto veniva eseguito direttamente sul componente, lasciando pochissimo margine di correzione ed un alto numero di insuccessi con relativo abbandono della realizzazione.
Il mio laboratorio è tappezzato di basette così realizzate ma mai terminate perché alla fine, da qualche parte, avevo dimenticato una connessione. Inoltre, bisogna anche dire che con le basette pre-forate, si era vincolati al passo tra i fori che queste offrivano, impedendo l'uso dei più sofisticati componenti SMD o logiche complesse o processori vari.

Gli anni sono passati, ho realizzato nel frat-

di Paolo D'Angella
p.dangella@infinito.it

Funziona!

tempo, pochissimi circuiti e quando oramai stremato, avendo capito che l'unico sistema sicuramente funzionante, elegante e pulito era quello con i trasferibili, mi sono deciso di riacquistare questi benedetti trasferibili ma sono incappato nella non disponibilità sul mercato.
Oramai nessuno più li usava e quei pochi che si trovavano erano ingialliti e non incollavano più per i lunghi anni di immagazzinamento subito.
Basta! dissi un giorno decidendo di impegnarmi nel trovare una soluzione piuttosto che cambiare hobby e dedicarmi alle bocce.
Incappai in un articolo che descriveva la possibilità di realizzare i circuiti stampati con la stampante bubble jet, ma anche li, i risultati non erano esattamente quelli voluti.
Spesso il risultato rassomigliava ad un quadro astratto di Picasso dopo essere caduto in una pozza di fango.
Non c'era nulla da fare, dovevo trovare io una soluzione al problema, applicarmi e risolvere ed il pensiero in quel frangente andava verso una fresa numerica assistita da un ottimo trapanino al alta velocità di rotazione realizzata grazie alla mia capacità di costruire anche componenti meccanici unitamente ad un controller con processore ma… lo stampato del controller come lo realizzo, con i trasferibili oppure con il pennarello indelebile?
Alla fine, decido di analizzare al fondo il processo xerografico a stampa laser e grazie alle mie conoscenze nel settore, avendoci lavorato per lungo tempo, sono riuscito a realizzare il mio primo prototipo di circuito stampato con tecnologia laser.
Non vi impressionate, non ho costruito nulla, mi sono semplicemente affidato ad una stampante laser ad alta risoluzione ed ho fatto fare tutto (quasi tutto) a lei.
Il processo, sfrutta la proprietà del Toner (inchiostro di stampa), di raggiungere la fusione a circa 180-200 gradi centigradi.

Figura 1 *Master dopo la realizzazione*

Il processo xerografico si sviluppa attorno ad un cilindro di materiale semiconduttore (selenio, solfuro di cadmio, metacrilato di stirene etc.) che è particolarmente sensibile alla illuminazione infatti, riduce la sua resistenza elettrica, portandola a zero, in presenza di una fonte di illuminazione.
Questo Cilindro, viene prima caricato con l'irraggiamento di una forte sorgente di cariche elettriche che si depositano sulla sua superficie che nello stato iniziale è praticamente nella sua condizione di massima resistenza elettrica poiché si trova al buio.
Successivamente, durante la sua rotazione, viene illuminato da un raggio laser o attra-

verso una fonte luminosa a LED che provoca, nel punto di illuminazione, la riduzione assoluta delle cariche elettriche disposte sulla sua superficie per effetto dell'abbattimento della sua resistenza elettrica.
Successivamente, sempre durante la sua rotazione, viene sottoposto alla nebulizzazione del toner, composto da materiale di trasporto caricato elettricamente di segno opposto alla carica deposta sul cilindro, da resine particolari e da pigmenti colorati (nel nostro caso di colore nero).
Il toner, si dispone sulla superficie del cilindro, solo dove vi è la presenza della carica elettrica di segno opposto, lasciando inalterata l'area che è stata precedentemente illuminata e quindi senza carica elettrica.

Nella fase successiva di rotazione, il toner verrà distaccato dalla superficie del cilindro, fornendo esternamente, una carica elettrica ancora maggiore a quella del cilindro stesso, proprio nel momento preciso in cui un foglio di carta viene fatto transitare tra il cilindro e questo generatore di alta carica elettrica.
Il risultato di questo processo, è la deposizione di una certa quantità di toner su un foglio bianco che lungo il suo percorso verso l'uscita della stampante o fotocopiatrice, verrà fatto passare in un forno di fusione (circa 180°), tra due rulli di pressione che cuoceranno letteralmente il toner sulla superficie della carta e per effetto della pressione, lo faranno penetrare nella primo strato superficiale della carta stessa.
L'idea è quella di riprodurre il processo nel verso opposto ovvero scaldare e togliere il toner fuso.

Infatti così ho fatto.

Ho realizzato il master, nel caso illustrato in figura 1 (una porzione), ho utilizzato un circuito preso da una vecchia rivista del 1982, ridisegnando completamente il master con un programma di grafica vettoriale.
L'ho stampato su foglio trasparente, adatto per fotocopiatrice ovvero in grado di sopportare le temperature di fusione dette in precedenza, con una stampante da 600dpi laser e non con una stampante a led poiché meno precisa nella stampa.
Ho sovrapposto il foglio lucido alla basetta ramata precedentemente pulita con acqua calda e VIM, facendo attenzione che la superficie a contatto fosse quella dove era stato deposto il toner e l'ho fissata con dello scotch da disegnatore.
A parte, ho fatto riscaldare il ferro da stiro nella posizione cotone/lana controllando che non emettesse vapore e con l'aiuto di un foglio di carta sottile interposto tra il ferro e

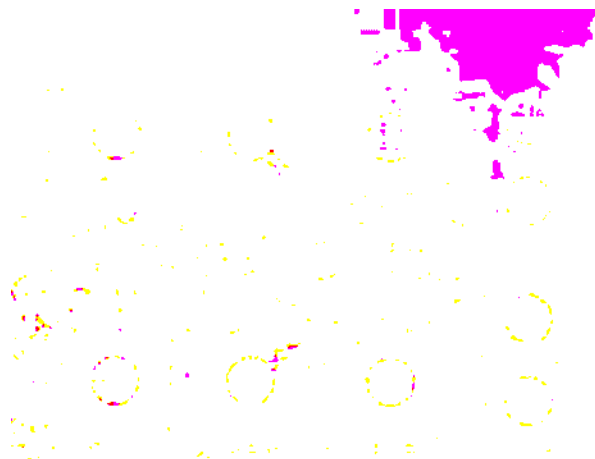



Figura 2 *Porzione della basetta di vetronite dopo la sovrapposizione per fusione del toner*

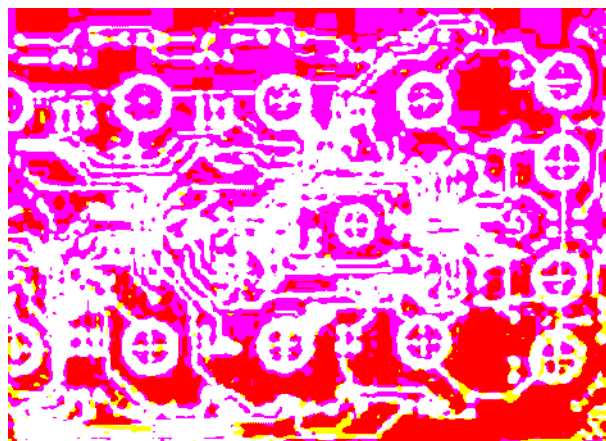



Figura 3 *Porzione della basetta di vetronite dopo l'aggressione del percloruro ferrico*

la basetta, ho stirato il tutto per circa 4 minuti, controllando che tutta la superficie fosse riscaldata omogeneamente ed associando anche una certa pressione.
Dopo circa 4 minuti di questo lavoro, ho tolto la pellicola trasparente ancora calda e magia delle magie, tutta l'immagine, si era trasferita sulla superficie di rame, come si può vedere dalla foto in figura 2.
Potrebbe capitare che in alcuni punti si sia insistito troppo con il ferro. In questo caso, si è portato il toner in ebollizione tanto che sulla superficie di rame si presenta un tratto a "buccia di arancio" che si rimuove facilmente e non è assolutamente solidale con la superficie di rame.
Se il tratto è di piccole dimensioni, si può intervenire con un pennarello indelebile altrimenti, si deve ripetere l'operazione.

Infine, la basetta così preparata va deposta in acido (percloruro ferrico) ottenendo quindi uno stampato "PERFETTO" figura 3.
Ultima cosa, il toner può essere rimosso con l'ausilio di una carta vetrata finissima, quella che usano i carrozzieri, immersa in acqua.

Buon lavoro!

MATERIALI OCCORRENTI

- Basetta ramata delle dimensioni idonee ad ospitare il master
- Stampante laser a 600 dpi
- Foglio lucido trasparente per la stampante o fotocopiatrice
- VIM o polvere abrasiva leggera
- Ferro da stiro
- Carta vetrata finissima

Riflettori su... COMUNICAZIONI BLUETOOTH

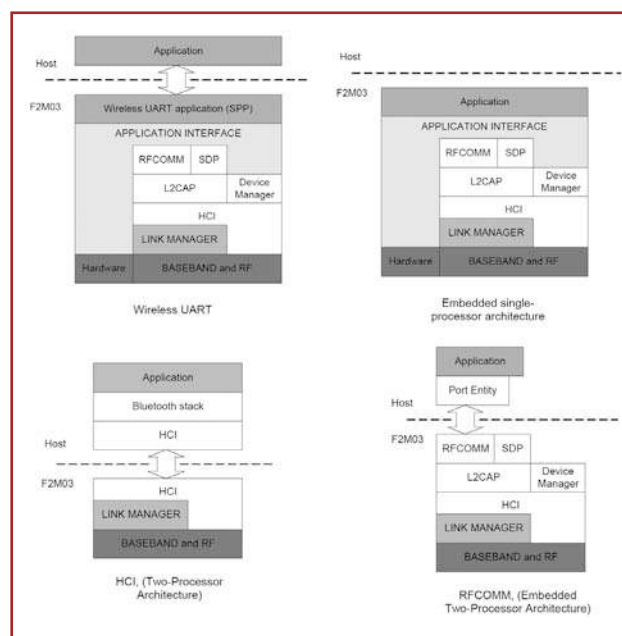
Una gamma di prodotti Bluetooth per soddisfare ogni esigenza di comunicazione: cable replacement RS232, HCI, RFCOMM e anche audio. Tutto questo in moduli di dimensioni ridottissime...

IL DONGLE

Pronto per l'uso, il dongle F2M01C1, una volta connesso alla porta seriale RS232 di qualsiasi apparecchiatura, è in grado di effettuare la comunicazione seriale via radio su 2,4GHz sfruttando il protocollo Bluetooth. Il vantaggio di utilizzare un protocollo standard come Bluetooth, permette di poter interagire con qualsiasi apparecchiatura Bluetooth quali Notebook, palmari, telefoni cellulari, o un secondo dongle remoto.

La configurazione è semplice ed avviene tramite l'opportuno software di configurazione fornito gratuitamente.
Il dispositivo è in classe 1, il che significa che il raggio di copertura può raggiungere i 100m se non vi sono ostacoli.

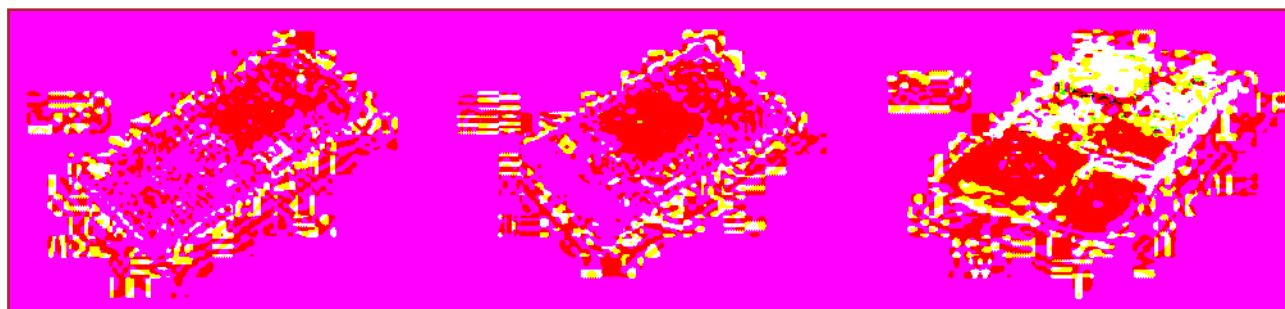
NON SOLO DONGLE

Per gli integratori più esigenti, Free2Move ha realizzato anche i moduli Bluetooth in versione SMD: tutto il necessario in moduli poco più grandi di un francobollo. I moduli sono disponibili in classe 1 e 2 (la classe 2 garantisce un raggio di copertura fino a 25m) e possono essere forniti con diversi tipi di firmware: UART, HCI, RFCOMM e Embedded.

Con il firmware UART è possibile implementare il cable replacement di connessioni RS232 senza l'ausilio di specifici driver. Nel caso di applicazioni con PC, può essere utile suddividere lo stack Bluetooth in due layer uno dei quali viene gestito direttamente dal PC. In questo caso si deve utilizzare il firmware HCI (Host Controller Interface). Per soluzioni embedded a singolo processore è possibile utilizzare il processore RISC interno ai moduli. In questo caso si possono implementare applicazioni single-chip come sistemi audio, gestione di mouse o testiere (HID) o altre particolari applicazioni custom. La versione RFCOMM permette infine di sviluppare applicazioni embedded basate su due processori, utili in tutti quei casi in cui l'host non ha risorse sufficienti per gestire l'intero stack Bluetooth.

Con i nuovi moduli Free2Move

APPLICAZIONI

I moduli Bluetooth Free2Move sono adatti a numerose applicazioni: domotica, remotizzazione di sensori, sistemi embedded, auricolari cordless, periferiche per PC (mouse, tastiere, dongle USB), telefonia ed applicazioni per palmari.

Per informazioni sui prodotti Free2Move:
Elettroshop
Via Cadorna, 27/31
20032 Cormano (MI)
Tel. 02.66504794 Fax 02.66508225
info@elettroshop.com – www.elettroshop.com

	F2M01C1	F2M03C1	F2M03C2	F2M03AC2
Antenna integrata	SI	NO	NO	SI
Classe	1	1	2	2
I/O digitali	-	8	8	8
I/O analogici	-	2	2	2
Dimensioni (mm)	48x34x19	33x14	25x14	24x13
Audio codec	NO	NO	NO	SI
Audio PCM	MO	SI	SI	SI
Firmware ver.	UART	UART, HCI, RFCOMM, Embedded	UART, HCI, RFCOMM, Embedded	UART, HCI, RFCOMM, Embedded
BT Compliant	v.1.1	v.1.1	v.1.1	v.1.1
Interfaccia USB	NO	SI	SI	SI

Riflettori su... grifo® QTP 03:

Potente ed economico Pannello Operatore dotato fino a 3 Tasti con la possibilità di gestire vari tipi di Display LCD o Fluorescenti.
Il QTP 03 può essere impiegato in abbinamento a qualsiasi CPU in grado di gestire una linea seriale, a livello TTL o RS232, oppure una linea I2C BUS.

Il **QTP 03** non è un banale display seriale, bensì un completo ed efficiente Pannello Operatore in grado di gestire, in maniera intelligente, tutta una serie di comandi che riguardano la rappresentazione dei messaggi; la comunicazione con il mondo esterno; la memorizzazione di messaggi; l'emissione di segnalazioni acustiche come il BELL, la gestione di tasti, eccetera.
Il QTP 03 può essere utilizzato nelle applicazioni che necessitano di rappresentazione messaggi e gestione sino a 3 differenti combinazioni di I/O come tasti od altro. Può essere alimentato a 5 Vdc ed a bordo scheda ha tutto ciò che serve per funzionare e per comunicare, tramite una linea a livello TTL, RS 232 oppure tramite I2C BUS.

SPERIMENTAZIONE

Grazie al suo basso prezzo il QTP 03 è l'ideale anche nel campo hobbistico e sperimentale. Il suo notevole numero di comandi, illustrati dettagliatamente nel manuale d'uso in Italiano, gli consentono di trovare delle soluzioni efficaci alle molteplici esigenze che si incontrano nelle varie applicazioni. Il manuale illustra un vasto numero di combinazioni di uso ed esempi che aiutano l'utente a sfruttare al meglio questo Pannello Operatore.

VASTA GAMMA DI DISPLAY SUPPORTATI

L'elettronica del QTP 03 ha un ingombro che è sempre inferiore all'ingombro della sezione display e ciò facilita enormemente il montaggio in quanto è possibile usare, per il fissaggio, la foratura prevista dallo stesso display.
Il **QTP 03** è disponibile con 4 display LCD o 4 display Fluorescenti che possono essere da 20 o 40 caratteri per 2 o 4 righe.

CONTENITORI PER PANNELLI OPERATORE

Tra le esigenze più sentite dagli utilizzatori c'è quella di alloggiare il proprio Pannello Operatore all'interno di un robusto contenitore. Una buona soluzione è data dai contenitori QTP 72144, per i display LCD e Fluorescenti,

Figura 1 Gli 8 tipi di display utilizzabili sui Panneli Operatore QTP 03

Quick Terminal Panel con 3 linee di I/O

da 20 caratteri per 2 righe e dal QTP 96192 per la versione Display 20x4 BIG.
I contenitori sono in alluminio ed hanno un livello di protezione sulla parte anteriore di IP54. Il pannello anteriore di alluminio del contenitore è già dotato della foratura per il montaggio dei 3 tasti quindi occorre solo ritagliare i fori sulla mascherina in policarbonato. Per il fissaggio si utilizzano i 4 alberi filettati, i relativi dadi, che combaciano perfettamente con la foratura presente sul display.

Figura 2 Vista posteriore, di un pannello QTP 72144, con montato un QTP 03-C2

CARATTERISTICHE RIASSUNTIVE GENERALI

Il QTP 03 indipendentemente dal display scelto è una soluzione molto efficiente, ed a basso costo, per dotare l'applicazione di un sistema di visualizzazione di messaggi.
La possibilità di poter richiamare, con una sequenza di comandi, una serie di messaggi memorizzati nella EEPROM di bordo è estremamente vantaggioso così come poter disporre di un potente **Buzzer**, per la funzione di **Bell**, e ciò fa aumentare le possibilità di utilizzo.
L'opportunità di poter gestire fino a 3 tasti, oppure, associare queste 3 linee per gestire dei messaggi attivabili dalle combinazione degli ingressi, proiettano il QTP 03 in un numero di utilizzi,solo fino ad ieri, impensabili.

Per ulteriori informazioni:
GRIFO®
Via dell'Artigiano, 8/6 - 40016
San Giorgio di Piano (BO)
Tel 051-892052 Fax 051-893661
www.grifo.it - www.grifo.com

Figura 3 Esempio di collegamento in rete di QTP, collegate in I2C BUS, e pilotate da una generica scheda di CPU

Figura 4 Display 20x4 caratteri, normale e BIG, e contenitore QTP 96192 con 3 tasti

Rispondi correttamente al quiz e potrai vincere un abbonamento omaggio a

Partecipare è semplicissimo:

rispondi al quesito seguendo il regolamento e, se la risposta si rivelerà esatta, potrai vincere un abbonamento omaggio (o il rinnovo qualora fossi già abbonato) a Fare Elettronica. Ogni mese sulle pagine della rivista troverai la soluzione del quesito del mese precedente e il nome del vincitore di uno dei quesiti pubblicati. Per tutti i partecipanti che risponderanno utilizzando l'apposito modulo su www.farelettronica.com/lesotutte, è previsto comunque un coupon del 10% di sconto utilizzabile per un acquisto sullo shop di www.farelettronica.com.

QUESITO codice LST23904

Un PIC16F84A operante in modalità XT con quarzo a 4MHz viene programmato con il seguente codice assembler:

```
PROCESSOR 16F84A
RADIX      DEC
include    "P16F84A.INC"

num        equ    11
ORG        0x00
movlw      num
clrc
rlf        W,0
rlf        W,1
addlw      0x0C
andlw      0x1C
rrf        W,0
end
```

Si chiede quale sia il contenuto dell'accumulatore W al termine del programma e in quanto tempo il programma stesso viene eseguito.

Scadenza: il termine ultimo per rispondere è il 31 Maggio 2005

Regolamento

1) Il quiz è aperto a tutti i lettori.
2) Saranno considerate esclusivamente le risposte pervenute entro la scadenza indicata nel quesito.
3) Inviare la risposta compilando il modulo su www.farelettronica.com/lesotutte oppure via **fax** allo 02/66508225 o tramite **lettera** a Inware Edizioni, via Cadorna 27- 20032 Cormano (MI), citando il codice del quesito.
4) Indicare nella risposta il tuo nome, cognome, indirizzo, telefono ed eventuale indirizzo email. Non ci sarà possibile considerare le risposte con **dati mancanti o insufficenti.**